

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UNA HERRAMIENTA WEB PARA LA ANOTACIÓN DE TRATAMIENTOS DE DATOS PERSONALES EN POLÍTICAS DE PRIVACIDAD EN ESPAÑOL

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INGENIERÍA EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

FÉLIX ANDRÉS MIÑO MORALES

DIRECTOR: MSc. GABRIEL ROBERTO LÓPEZ FONSECA

CODIRECTOR: MSc. FRANKLIN LEONEL SÁNCHEZ CATOTA

Quito, enero 2021

AVAL

Certificamos que el presente trabajo fue desarrollado por Félix Andrés Miño Morales, bajo nuestra supervisión.

MSc. GABRIEL ROBERTO LÓPEZ FONSECA
DIRECTOR DEL TRABAJO DE TITULACIÓN

MSc. FRANKLIN LEONEL SANCHEZ CATOTA
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Félix Andrés Miño Morales, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

FÉLIX ANDRÉS MIÑO MORALES

DEDICATORIA

Para mis padres que sin su guía nada de esto sería posible.

AGRADECIMIENTO

Agradezco a Dios.

A mis padres por sus sacrificios y enseñanzas para formar la persona que soy.

A mi compañera de vida, Emily, por todo su amor y apoyo incondicional a lo largo de estos años.

A los amigos que me dejó esta etapa, Diego, Jorge, Israel y Alexander, con los cuales he compartido gran parte de mi carrera universitaria y de mi vida.

A esos amigos que llegan inesperadamente, Sergio, Alexis, Alex y Gorky, quienes me enseñaron a amar aún más la programación.

A mis tutores, Msc. Gabriel López y Msc. Franklin Sánchez, por su guía.

Un especial agradecimiento a Danny Guamán por la confianza que tuvo en mí para ser parte de este proyecto, por todo el tiempo invertido y la ayuda prestada, gracias Danny.

A todos mis profesores que de una u otra manera han influido en mi formación.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	X
ÍNDICE DE CÓDIGOS	XI
RESUMEN	XII
ABSTRACT	XIII
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	3
1.2. ALCANCE	3
1.3. MARCO TEÓRICO	8
1.3.1. DEFINICIONES.....	9
1.3.2. PRIVACIDAD Y PROTECCIÓN DE DATOS PERSONALES	9
1.3.3. PROYECTO DE LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES	14
1.3.4. PLODPD VS. RGPD.....	20
1.3.5. CATEGORÍAS DE TRATAMIENTO DE DATOS PERSONALES	24
1.3.6. METODOLOGÍA SCRUM.....	26
1.3.7. HERRAMIENTAS Y LENGUAJES DE PROGRAMACIÓN UTILIZADOS	27
1.4. RELACIÓN CON TRABAJOS SIMILARES EN EL ÁREA.....	30
2. METODOLOGÍA.....	31
2.1. METODOLOGÍA DE DESARROLLO	31
2.2. ANÁLISIS.....	32
2.2.1. REQUERIMIENTOS FUNCIONALES.....	32
2.2.2. REQUERIMIENTOS NO FUNCIONALES.....	34
2.2.3. DIAGRAMA DE CASOS DE USO	34
2.2.4. HISTORIAS DE USUARIO	36
2.2.5. PRODUCT BACKLOG	38

2.2.6. PLANIFICACIÓN DE <i>SPRINTS</i>	38
2.3. DISEÑO	41
2.3.1. ESTRUCTURA DE UNA ANOTACIÓN.....	42
2.3.2. MODELO – DISEÑO DE BASE DE DATOS.....	43
2.3.3. VISTA – DISEÑO DE INTERFACES	47
2.3.4. CONTROLADOR – DISEÑO DE CLASES	49
2.4. IMPLEMENTACIÓN.....	56
2.4.1. MÓDULO DE ADMINISTRACIÓN - <i>SPRINT 1</i>	58
2.4.2. MÓDULO DE ANOTACIÓN - <i>SPRINT 2</i>	69
2.4.3. MÓDULO DE CONSOLIDACIÓN - <i>SPRINT 3</i>	72
2.4.4. MÓDULO DE VISUALIZACIÓN - <i>SPRINT 4</i>	78
3. VERIFICACIÓN Y VALIDACIÓN	83
3.1. PRUEBAS FUNCIONALES	84
3.1.1. MÓDULO DE ADMINISTRACIÓN Y AUTENTICACIÓN - <i>SPRINT 1</i>	85
3.1.2. MÓDULO DE ANOTACIÓN, CONSOLIDACIÓN Y VISUALIZACIÓN - <i>SPRINT 2, 3 y 4</i>	95
3.2. VALIDACIÓN	113
3.2.1. FUNCIONAMIENTO GENERAL DE LA HERRAMIENTA	113
3.2.2. ENCUESTA DE SATISFACCIÓN REQUISITOS FUNCIONALES	114
3.2.3. ENCUESTA DE SATISFACCIÓN REQUISITOS NO FUNCIONALES	114
3.2.4. USO DE LA HERRAMIENTA EN PROCESOS DE INVESTIGACIÓN	115
3.3. RESULTADOS Y DISCUSIÓN.....	115
3.3.1. FUNCIONAMIENTO GENERAL DE LA HERRAMIENTA	115
3.3.2. ENCUESTA DE SATISFACCIÓN REQUISITOS FUNCIONALES	119
3.3.3. ENCUESTA DE SATISFACCIÓN REQUISITOS NO FUNCIONALES	122
4. CONCLUSIONES Y RECOMENDACIONES.....	123
4.1. CONCLUSIONES	123
4.2. RECOMENDACIONES	125
4.2.1. RECOMENDACIONES OBTENIDAS DE LA ENCUESTA DE SATISFACCIÓN DE REQUISITOS FUNCIONALES.....	125
4.2.2. RECOMENDACIONES DEL EQUIPO DE DESARROLLO.....	126
5. REFERENCIAS BIBLIOGRÁFICAS	127
6. ANEXOS.....	130

ÍNDICE DE FIGURAS

Figura 1.1 Resumen funcionalidad herramienta	3
Figura 1.2 Diagrama de despliegue de la herramienta	5
Figura 2.1 Diagrama de casos de uso de la herramienta de anotación	35
Figura 2.2 Diagrama de casos de uso herramienta de visualización	35
Figura 2.3 Historia de usuario HU-01 en Trello	37
Figura 2.4 <i>Sprint</i> 1 en Trello	41
Figura 2.5 Diagrama general de la herramienta	42
Figura 2.6 Estructura de una anotación.....	42
Figura 2.7 Diagrama entidad-relación	44
Figura 2.8 Diagrama relacional	45
Figura 2.9 Interfaz de consolidación (párrafo/sección inconsistente).....	48
Figura 2.10 Interfaz de anotación.....	49
Figura 2.11 Interfaz de visualización de política de privacidad anotada	49
Figura 2.12 Diagrama de clases de la herramienta	50
Figura 2.13 Diagrama de flujo de notificación de inconsistencia en anotación	52
Figura 2.14 Diagrama de flujo de cálculo de coeficiente inter-anotador.....	53
Figura 2.15 Diagrama de flujo notificación de sección inconsistente en consolidación....	54
Figura 2.16 Diagrama de flujo de presentación política de privacidad anotada	55
Figura 2.17 Diagrama de flujo filtrado de anotaciones en módulo visualización	56
Figura 2.18 Comunicación de componentes de la herramienta.....	57
Figura 2.19 Estructura del código de la herramienta	58
Figura 2.20 Componente Usuario	64
Figura 2.21 Resultado de: Código 2.17 y Código 2.18	69
Figura 2.22 Componentes de la página de anotación	70
Figura 2.23 Matriz de coincidencia binaria Alpha de Krippendorff	75
Figura 2.24 Lista de Tratamientos presentados al usuario para filtrado	80
Figura 2.25 Interfaz de filtrado a nivel de atributo (opción de limpieza de filtros).....	80
Figura 3.1 Diagrama de despliegue en AWS	83
Figura 3.2 Formulario de inicio de sesión.....	85
Figura 3.3 Pantalla de bienvenida herramienta anotación	86
Figura 3.4 Estructura de una petición y creación de un módulo a través de <i>Postman</i>	87

Figura 3.5 Consulta de módulo a través de <i>Postman</i>	87
Figura 3.6 Modificación de módulo a través de <i>Postman</i>	88
Figura 3.7 Eliminación de módulo a través de <i>Postman</i>	88
Figura 3.8 Formulario creación de usuario	89
Figura 3.9 Correo recibido al ser registrado como anotador de la herramienta.	89
Figura 3.10 Lista de usuarios en la herramienta.....	90
Figura 3.11 Formulario de edición de usuario	90
Figura 3.12 Lista de usuarios luego de edición	91
Figura 3.13 Intento de acceso con usuario inactivo.....	91
Figura 3.14 Prueba de eliminación de usuario manteniendo integridad de clave foránea	91
Figura 3.15 Prueba de eliminación de usuario	92
Figura 3.16 Formulario para creación de tratamiento y asignación de color.....	93
Figura 3.17 Lista de tratamientos.....	93
Figura 3.18 Prueba de edición de tratamiento.....	93
Figura 3.19 Lista de tratamientos luego de edición	94
Figura 3.20 Creación de atributo para prueba de integridad referencial	94
Figura 3.21 Prueba de eliminación de tratamiento que contiene atributos.....	94
Figura 3.22 Prueba de eliminación tratamiento que no contiene atributos.....	94
Figura 3.23 Lista de tratamientos para prueba de filtrado	95
Figura 3.24 Lista de tratamientos filtrada	95
Figura 3.25 Estructura del caso de prueba para pruebas funcionales de anotación	96
Figura 3.26 Lista de políticas de privacidad por anotar	98
Figura 3.27 Pantalla de anotación.....	99
Figura 3.28 Creación de tratamiento de prueba	99
Figura 3.29 Lista de tratamientos disponibles después de creación de tratamiento	100
Figura 3.30 Creación de atributo de prueba	100
Figura 3.31 Lista de tratamientos disponibles después de creación de atributo	100
Figura 3.32 Creación de valor de prueba	100
Figura 3.33 Lista de tratamientos disponibles después de creación de valor de prueba	101
Figura 3.34 Creación de Anotación 1.1	101
Figura 3.35 Registro de anotación 1.1 en la base de datos.....	101
Figura 3.36 Notificación de finalización de política de privacidad	102
Figura 3.37 Conteo de anotaciones sobre una sección.....	102
Figura 3.38 Diálogo de edición de anotación	103
Figura 3.39 Resumen de anotaciones realizadas sobre una sección	103
Figura 3.40 Diálogo eliminación de anotación.....	103

Figura 3.41 Notificación de inconsistencia con tratamientos sugeridos	104
Figura 3.42 Notificación de inconsistencia sin tratamientos sugeridos	104
Figura 3.43 Lista de políticas de privacidad lista para consolidar	105
Figura 3.44 Registro de usuarios asignados para anotar "Política de privacidad 1"	105
Figura 3.45 Registros de la base de datos de políticas de privacidad	106
Figura 3.46 Detalles de "Política de privacidad 1"	106
Figura 3.47 Resumen de anotaciones sección 1	107
Figura 3.48 Copia de anotación en etapa de consolidación	107
Figura 3.49 Resumen de anotaciones en etapa de consolidación	107
Figura 3.50 Sección 2 de "Política de privacidad 1", consistente sin anotaciones.....	108
Figura 3.51 Sección 1 de "Política de privacidad 1", consistente con anotaciones.....	108
Figura 3.52 Sección 3 de "Política de privacidad 1", inconsistente.....	108
Figura 3.53 Página de bienvenida herramienta de visualización	109
Figura 3.54 Políticas de privacidad disponibles para visualización.....	109
Figura 3.55 Ejemplo de filtrado de lista de políticas de privacidad.....	110
Figura 3.56 Página de visualización de política de privacidad.....	110
Figura 3.57 <i>Tooltip</i> con detalles de anotación	111
Figura 3.58 Tratamiento con atributos desplegados.....	111
Figura 3.59 Anotaciones de filtradas por tratamiento "Recolección".....	112
Figura 3.60 Anotaciones filtradas por tratamiento "Recolección" y atributo "Tipo de dato"	113
Figura 3.61 Resumen de políticas de privacidad para fase de validación.....	114
Figura 3.62 Secciones totales por política de privacidad	116
Figura 3.63 Número de palabras por política de privacidad.....	117
Figura 3.64 Total de anotaciones por política de privacidad.....	117
Figura 3.65 Número de anotaciones por tratamiento de datos (Gráfico de barras)	118
Figura 3.66 Número de anotaciones por tratamiento de datos por política de privacidad	119

ÍNDICE DE TABLAS

Tabla 1.1 Contrastación de definiciones PLOPDP vs RGPD	21
Tabla 1.2 Comparativa de principios de protección PLOPDP vs. RGPD	23
Tabla 1.3 Resumen de tratamientos de datos	26
Tabla 2.1 Miembros del equipo Scrum	32
Tabla 2.2 Formato de una historia de usuario	36
Tabla 2.3 Historia de usuario HU-01: Inicio y cierre de sesión	37
Tabla 2.4 <i>Product Backlog</i>	38
Tabla 2.5 Planificación de <i>sprints</i>	40
Tabla 2.6 Lenguajes de programación y herramientas usadas para la implementación ..	57
Tabla 2.7 Matriz de datos de confiabilidad binaria para tres codificadores y diez unidades de análisis	75
Tabla 3.1 Direcciones web de la herramienta	84
Tabla 3.2 Resumen de cumplimiento de las historias de usuario	84
Tabla 3.3 Anotaciones realizadas sobre políticas de privacidad en caso de prueba	96
Tabla 3.4 Preguntas de validación requerimientos no funcionales	115
Tabla 3.5 Lista final de tratamientos, atributos y valores	116
Tabla 3.6 Resultados encuesta de satisfacción de requisitos funcionales	119
Tabla 3.7 Encuesta de satisfacción de requisitos no funcionales respondida	122

ÍNDICE DE CÓDIGOS

Código 2.1 Consulta ejecutada con SQL y Flask-SQLAlchemy	47
Código 2.2 Clase Usuario	59
Código 2.3 Función Guardar Usuario.....	60
Código 2.4 Función respuesta	60
Código 2.5 Función obtener Usuario.....	61
Código 2.6 Función editar Usuario.....	61
Código 2.7 Función eliminar Usuario	62
Código 2.8 <i>Endpoint</i> para CRUD de Usuario	62
Código 2.9 Registro de <i>paths</i> de <i>endpoints</i> del <i>backend</i>	63
Código 2.10 Clase <code>UsuarioDTO</code>	64
Código 2.11 Componente usuario 1 de 4.....	65
Código 2.12 Componente usuario 2 de 4.....	66
Código 2.13 Componente usuario 3 de 4.....	66
Código 2.14 Componente Usuario 4 de 4	66
Código 2.15 Servicio de Usuario.....	67
Código 2.16 Interfaz <code>UsuarioConsultar</code>	67
Código 2.17 Código HTML de interfaz de usuario 1/2.....	68
Código 2.18 Código HTML de interfaz de usuario 2/2.....	68
Código 2.19 Servicio para comunicación de valores entre componentes.....	71
Código 2.20 Ejemplo de suscripción a un <code>BehaviorSubject</code>	71
Código 2.21 Función para selección de texto	72
Código 2.22 Cálculo de coeficiente inter-annotador 1 de 3	76
Código 2.23 Cálculo de coeficiente inter-annotador 2 de 3	77
Código 2.24 Cálculo de coeficiente inter-annotador 3 de 3	77
Código 2.25 Función <code>politicaOriginal</code> (Presentación de política sin filtros).....	79
Código 2.26 Función <code>darEstiloTitulo</code> y <code>darEstiloAnotacion</code>	79
Código 2.27 Función para manejo de filtro de tratamiento de datos.....	81
Código 2.28 Código HTML para filtrado de anotaciones por tratamiento de datos.....	82
Código 2.29 Función para filtrar anotaciones por tratamiento de datos.....	82

RESUMEN

Este proyecto de titulación presenta el desarrollo de una herramienta web para la anotación de tratamientos de datos expresados en políticas de privacidad en español.

La herramienta se divide en dos componentes web, los cuales fueron desarrollados con el lenguaje de programación TypeScript y el *framework* Angular:

- Herramienta de anotación: permite realizar las anotaciones de los tratamientos de datos personales expresados en las políticas de privacidad.
- Herramienta de visualización: permite visualizar las anotaciones realizadas en la herramienta de anotación de manera estructurada.

Ambos componentes web son soportados por una misma aplicación de servidor, el cual fue desarrollado con el lenguaje de programación Python y el *framework* Flask.

Este documento está organizado así:

En el primer capítulo se presenta el marco teórico, el cual aborda: la privacidad y la protección de datos, el Proyecto de Ley Orgánica de Protección de Datos Personales (PLOPDP) ecuatoriano, la definición de categorías de tratamientos de datos a partir del PLOPDP, una breve comparación del PLOPDP con el RGPD, la metodología ágil Scrum y las herramientas y lenguajes de programación usados para el desarrollo de la herramienta.

En el segundo capítulo se presenta la metodología usada para el desarrollo de la herramienta, la cual está basada en Scrum. El proceso de desarrollo se llevó a cabo a través de 3 fases: análisis de requerimientos, diseño de la herramienta e implementación, las tres fases se describen en este capítulo.

En el tercer capítulo se presenta la evidencia de las pruebas de funcionalidad de las herramientas de anotación y visualización. También se presentan encuestas de satisfacción que sirven como validación de este trabajo.

Finalmente, en el cuarto capítulo se presentan las conclusiones obtenidas una vez finalizada la herramienta, así como recomendaciones para trabajos futuros que se realicen sobre la herramienta.

PALABRAS CLAVE: privacidad, protección de datos personales, tratamiento de datos personales, anotación de tratamientos de datos personales, políticas de privacidad, PLOPDP, Scrum, Python, Flask, TypeScript, Angular.

ABSTRACT

This graduation Project presents the development of a web tool for the annotation of data processing stated in privacy policies in Spanish.

The tool is divided in two web components, which were developed with the TypeScript programming language and the Angular framework:

- Annotation tool: lets on the annotation of data processing stated on privacy policies.
- Visualization tool: allows the view of the annotations performed in the annotation tool in a structured way.

Both web components are supported by a server application which was developed using the Python programming language and the Flask framework.

This document is organized as follows:

The first chapter presents the theoretical framework, which covers: privacy and data protection, the *Proyecto de Ley Orgánica de Protección de Datos Personales* (PLOPDP), the definition of data processing categories based on PLOPDP, a brief comparison between the PLOPDP and GDPR, the agile methodology Scrum and the tools and programming languages used in the development of this project.

The second chapter presents the methodology followed for the development of the tool, which is based on Scrum. The development process was performed across three phases: requirement's analysis, design of the tool and implementation. All three phases are explained in this chapter.

The third chapter presents the evidence of the functional tests carried out in the annotation and visualization tools. Satisfaction surveys are also presented as a way of validating the tool.

Finally, in chapter four the conclusions obtained by the development of the tool are presented, as well as recommendations that could be performed in future associated work.

KEYWORDS: privacy, data protection, data processing, annotation of data processing, privacy policies, PLOPDP, Scrum, Python, Flask, TypeScript, Angular.

1. INTRODUCCIÓN

El ritmo acelerado del desarrollo tecnológico, particularmente de aplicaciones móviles y web, permiten simplificar tareas de la vida cotidiana y ofrecen mayores comodidades a sus usuarios; sin embargo, esto puede conllevar una recolección masiva de datos.

Las aplicaciones móviles constituyen un ejemplo interesante de tecnologías que recolectan datos personales porque, además de la información que se almacena en el dispositivo móvil, estos cuentan con un amplio rango de sensores integrados capaces de generar datos por sí mismos. [1]

Muchos casos de filtraciones de datos personales y la implicada violación a la privacidad son conocidos a nivel mundial. Por ejemplo, Cambridge Analytica [2] hizo uso de datos provistos por los usuarios para influir en su comportamiento en dos casos particulares: 1) su intervención en la campaña presidencial de Trump en 2016, la cual la empresa describe como “*decisiva*” para el triunfo del candidato republicano y 2) su trabajo en la campaña de Leave.EU uno de los grupos a favor del Brexit en Reino Unido donde finalmente ganaría el “*sí*”.

Algunos estados regulan el tratamiento de datos personales a través de la expedición de normativas. Uno de los más recientes es el Reglamento General de Protección de Datos Personales (RGPD) emitido en la Unión Europea y con vigor desde el 25 de Mayo de 2018 [3]. Así también, Ecuador realizó sus primeros esfuerzos. El 19 de septiembre de 2019 a través del Ministerio de Telecomunicaciones y Sociedad de la Información (MINTEL) entregó a la Asamblea Nacional el Proyecto de Ley Orgánica de Protección de Datos Personales (PLOPDP) [4] que busca regular el tratamiento de datos a través de la definición de un conjunto de principios de protección y derechos de los usuarios.¹

El PLOPDP define las políticas de privacidad como: “*documento físico, electrónico o en cualquier formato generado por el responsable del tratamiento de datos personales que debe obligatoriamente ponerse a disposición del titular, a partir del momento en el cual se recaben sus datos personales y debe estar disponible de forma permanente, con el objeto de garantizar el derecho a la transparencia*” [4]. Como menciona Carmela Troncoso, una de las principales expertas en el ámbito de la privacidad y protección de datos, las políticas de privacidad son el principal medio usado para proveer transparencia al usuario respecto al tratamiento de sus datos personales [5]. Si bien el estado de la técnica muestra que las

¹ La Ley Orgánica de Protección de Datos Personales fue presentada como proyecto de ley, a la fecha de presentación de este trabajo aún no ha sido aprobada. En este proyecto se hará uso de la última versión disponible.

políticas de privacidad son el principal medio usado para proveer de transparencia, éstas presentan varias limitaciones. Entre las que podemos resaltar: 1) los usuarios tienen problemas con su interpretación lo que deriva a que sean aceptadas sin apenas haber sido leídas, debido a su complejidad y extensión [6] y 2) al estar escritas en lenguaje natural son una barrera en la automatización de la anotación de tratamientos de datos personales, por lo que esta acción debe hacerse de forma manual [1].

Como ayuda para la interpretación de las políticas de privacidad algunos trabajos proponen realizar anotaciones de los tratamientos de datos personales que son expresados en ellas: 1) de forma manual [7][8] , o 2) de forma automática con sistemas de inteligencia artificial [9][10][11][12]. Ahora, si bien se han realizado varias contribuciones en este ámbito, estas se concentran principalmente en políticas de privacidad en inglés, existiendo un escaso, casi nulo, trabajo en políticas de privacidad en español. Además, estas contribuciones abordan la privacidad tomando como referencia la regulación de sus propias jurisdicciones. El PLOPDP en el Ecuador, pese a estar basado en el RGPD, posee características propias que requieren ser analizadas. Finalmente, es importante mencionar que todas estas herramientas no están disponibles para ser usadas.

Este trabajo se enfoca en sentar las bases para superar estas limitaciones. Mediante la creación de una herramienta web que permita:

1. La anotación manual de tratamientos de datos personales por parte de varios anotadores de manera concurrente.
2. La definición de nuevos tratamientos de datos.
3. El cálculo de coeficiente inter-anotador para mostrar el nivel de fiabilidad de las anotaciones de varios anotadores.
4. La presentación de anotaciones en una forma estructura y sencilla.

Además de las características previas, durante el desarrollo de este trabajo se realiza la anotación de ciertos tratamientos de datos de acuerdo con el PLOPDP como una manera de validar la herramienta desarrollada. Finalmente, es importante mencionar que este trabajo de titulación sirvió como herramienta de soporte de la investigación “GDPR Compliance Assessment for Cross-border Personal Data Transfers in Android Apps” [13], cuyo Corpus se encuentra disponible en <https://data.mendeley.com/datasets/ws6cx9p65d/1>.

1.1. OBJETIVOS

El objetivo general de este proyecto de titulación es:

- Desarrollar una herramienta web para realizar anotaciones de los tratamientos de datos personales que se informan a través de políticas de privacidad en español.

Los objetivos específicos de este proyecto son:

- Investigar los fundamentos teóricos necesarios para el desarrollo de la herramienta.
- Diseñar los módulos que conforman la herramienta.
- Implementar los módulos que hacen parte de la herramienta.
- Validar el funcionamiento de la herramienta.

1.2. ALCANCE

La Figura 1.1 muestra de manera resumida la funcionalidad de la herramienta de anotación. Como entrada de la herramienta se tiene una política de privacidad en español que, luego de ser anotada y seguir el respectivo proceso, podrá ser visualizada por un usuario. Esta herramienta servirá de soporte para la anotación manual de tratamientos de datos personales presentes en políticas de privacidad en general, aunque en este proyecto nos centramos en las políticas de privacidad en español. Particularmente, la herramienta permitirá realizar anotaciones de los tratamientos de datos personales expresados en el PLOPDP.

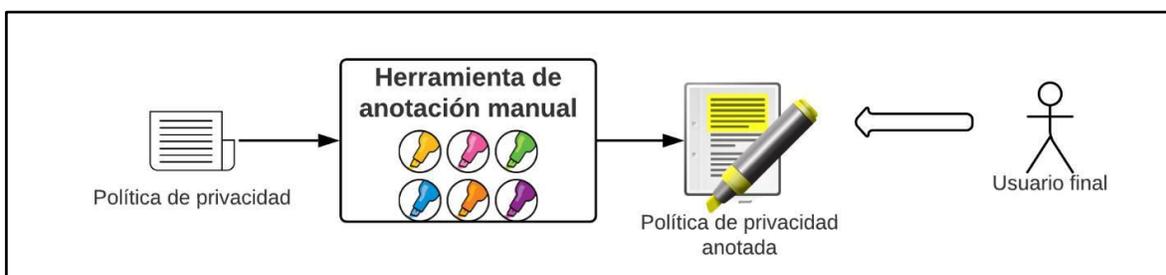


Figura 1.1 Resumen funcionalidad herramienta

Como se ilustra en la Figura 1.2, la herramienta propuesta estará compuesta por cuatro módulos principales:

- Módulo de administración.
- Módulo de anotación.
- Módulo de consolidación.

- Módulo de visualización.

Estos módulos serán alojados en dos herramientas web:

- Una herramienta web privada, cuyo acceso requerirá autenticación y contendrá:
 - Módulo de administración.
 - Módulo de anotación.
 - Módulo de consolidación.
- Y otro sitio web público, que no requerirá autenticación, y servirá para visualizar las políticas anotadas, y contendrá al:
 - Módulo de visualización.

La herramienta propuesta soportará dos tipos de usuarios: autenticados y no autenticados. Los usuarios autenticados se encontrarán en la herramienta web privada y podrán tomar uno de los dos tipos de roles que dispondrá la herramienta web:

- Rol administrador
- Rol anotador.

Por un lado, el rol administrador tendrá acceso a los siguientes módulos:

- Módulo de administración
- Módulo de anotación
- Módulo de consolidación

Mientras que el rol anotador solamente tendrá acceso a:

- Módulo de anotación.

Los usuarios no autenticados tendrán acceso solamente al sitio web público y serán mencionados en este documento como usuarios finales.

La herramienta soportará el acceso concurrente de múltiples usuarios autenticados o no autenticados, de manera que usuarios con rol anotador o rol administrador puedan trabajar sobre la misma política de privacidad y usuarios finales puedan observar las anotaciones realizadas sobre estas políticas de privacidad.

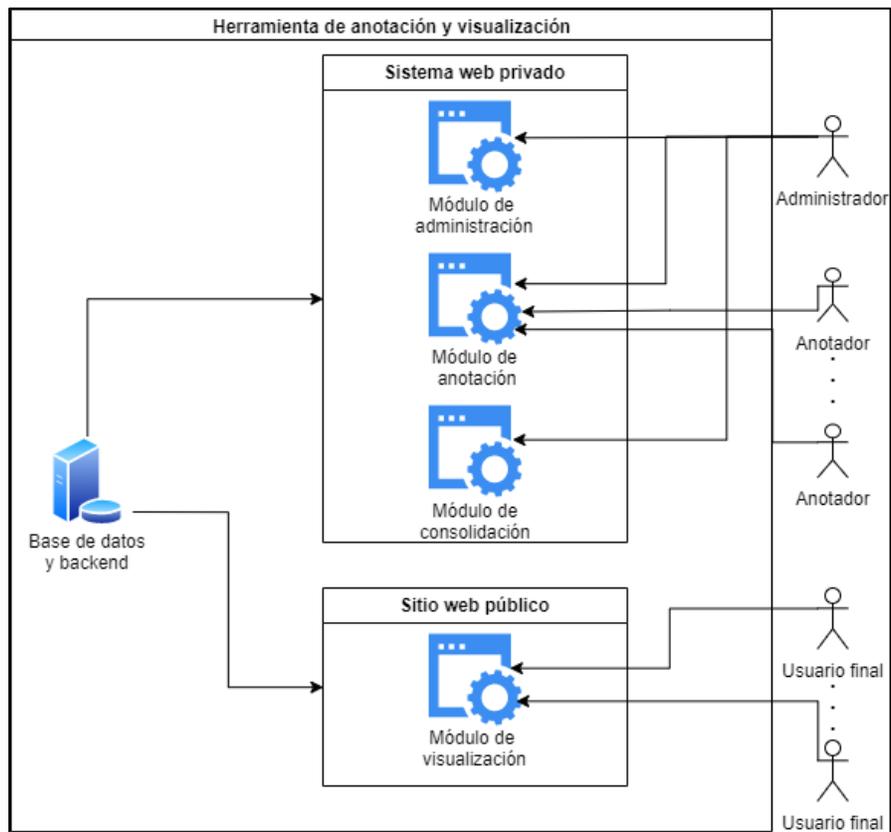


Figura 1.2 Diagrama de despliegue de la herramienta

A continuación, se describe la funcionalidad que tendrá cada módulo:

1. **Módulo de administración:**

Este módulo permitirá al usuario con rol administrador definir elementos necesarios para llevar a cabo las anotaciones. Concretamente, el módulo de administración incluirá los submódulos y funcionalidad indicada a continuación:

a. **Submódulo de administración de usuarios**

- Se podrá crear, editar, visualizar y eliminar usuarios.
- Se podrá definir el rol de los usuarios al momento de crearlos: administrador o anotador.

b. **Submódulo de administración de políticas**

- Este submódulo permitirá la creación (subida), edición, visualización y eliminación de políticas de privacidad.

- Se podrá cargar una política de privacidad para que sea anotada. El archivo deberá estar en formato `txt` y deberá cumplir ciertos parámetros para que pueda ser procesada de manera correcta.
- Se realizará un preprocesamiento del texto de la política de privacidad, de modo que se muestre cada párrafo o sección, de esta manera se facilitará el trabajo para el usuario que anotará la política.
- Se permitirá asignar una política de privacidad para anotación a un conjunto de usuarios con rol anotador o rol administrador (mínimo 2).

c. Submódulo de administración de tratamientos

- Permitirá la creación, edición, visualización y eliminación de tratamientos de datos personales. Estos tratamientos serán obtenidos del PLOPDP. Algunos ejemplos de tratamientos de datos son: recolección y difusión.
- Permitirá asignar a cada tratamiento una etiqueta (color) para facilitar su visualización al realizar anotaciones y al visualizar la política.

d. Submódulo de administración de atributos

- Se permitirá la creación, edición, visualización y eliminación de atributos de un tratamiento de datos. Por ejemplo, para el tratamiento de recolección le podrían corresponder los siguientes atributos: tipo de dato recolectados, propósito de recolección, entre otros.

e. Submódulo de administración de valores

- Se permitirá la creación, edición, visualización y eliminación de los valores que podría tomar un atributo. Por ejemplo, para el atributo tipo de dato recolectado sus valores podrían ser: localización, cookies, información financiera, información del dispositivo entre otras.
- No corresponde al análisis teórico de este trabajo el determinar los valores que corresponden a los atributos ya que representa un trabajo complejo e involucraría más tiempo del programado para la realización de este trabajo. El único valor que tendrán los atributos de cualquier tratamiento de datos será “genérico”.

2. Módulo de Anotación

Este módulo permitirá a los usuarios con rol anotador o rol administrador realizar y editar anotaciones de tratamientos de datos sobre las políticas de privacidad que les fueron previamente asignadas. Solo se podrán editar anotaciones realizadas por el mismo usuario, en ningún caso anotaciones realizadas por otro. Al usuario se le presentarán las políticas de privacidad que le fueron asignadas y que aún no fueron terminadas de anotar, también podrá ver su progreso en cada política.

Al seleccionar una política será presentada a nivel de párrafo o sección. Para realizar una anotación será necesario en primer lugar seleccionar un tratamiento de datos, para luego escoger un atributo que pertenece a este tratamiento y finalmente seleccionar el valor "genérico". Una vez seleccionado el valor se podrá subrayar el texto, será posible ingresar un comentario por cada anotación realizada y sólo podrá ser visualizada por el administrador. Al presionar guardar la anotación será enviada a la base de datos.

Cuando un usuario con rol anotador o rol administrador esté realizando la anotación de tratamientos y presione el botón guardar, en caso de existir conflictos con anotaciones hechas por otros usuarios con rol anotador o rol administrador sobre la misma sección se desplegará una advertencia de que no existe conformidad con la anotación realizada, sin embargo, el usuario que está realizando la anotación será quien tome la decisión de guardar o corregir las anotaciones previamente realizadas. Esta característica podrá ser activada o desactivada por el rol administrador.

3. Módulo de consolidación

Este módulo permitirá al usuario con rol administrador consolidar una política de privacidad previamente anotada por dos o más usuarios con rol anotador o rol administrador, es decir, en base a estas anotaciones previas fijar una anotación final que represente al conjunto de usuarios. El usuario con rol administrador podrá revisar los comentarios realizados por los usuarios con rol anotador o rol administrador y realizar un comentario que será el que el usuario final visualiza. Para poder empezar con la consolidación todos los usuarios asignados a realizar la anotación de dicha política deben haber concluido.

Así mismo, este módulo calculará el nivel de concordancia inter-anotador, que permitirá conocer el grado de similitud que existe en las anotaciones realizadas por los usuarios con rol anotador o rol administrador sobre la política de privacidad.

4. Módulo de visualización

Este módulo será público ya que su objetivo es que cualquier persona puede ingresar a la herramienta y visualizar las políticas de privacidad anotadas.

Las anotaciones de las políticas de privacidad y los comentarios que se presentarán en este módulo serán las realizadas por el usuario con rol administrador en la etapa de consolidación.

El usuario final podrá filtrar las anotaciones por el tratamiento de datos y por alguno de los atributos que componen este tratamiento. No se podrá filtrar dos tratamientos ni dos atributos de manera simultánea. El filtrado a nivel multi-atributo está fuera del alcance de este trabajo. Tampoco se podrá filtrar a nivel de valor.

Se usará la arquitectura Modelo Vista Controlador (MVC). Como lenguaje de programación para el controlador se usará Python junto con flask. Como entorno de desarrollo para el *backend* de la herramienta se utilizará PyCharm. Como motor de base de datos (Modelo) se tendrá MySQL y para gestionar el motor se usará MySQL Workbench. Además, se utilizará flask-SQLAlchemy como mapeador objeto-relacional. Para el desarrollo de la parte visual (Vista) de la herramienta se usará Angular junto con el editor de código Visual Studio Code.

Para el desarrollo de la herramienta y con la finalidad de presentar un desarrollo ordenado este trabajo tendrá como guía de metodología ágil a SCRUM.

La herramienta será validada a través de pruebas de funcionamiento, donde se comprobará el correcto desempeño de sus funcionalidades.

1.3. MARCO TEÓRICO

En esta sección se presentan algunos conceptos relacionados a la privacidad y protección de datos, que son necesarios para entender el contexto del presente trabajo. Se abordan tres perspectivas desde las cuales se puede estudiar la privacidad y protección de datos desde el ámbito de la ingeniería, según Troncoso et al [5]. Luego, se presenta el estudio de los principios de protección de datos expuestos en el PLOPDP, se realiza una breve contrastación del PLOPDP ecuatoriana con la regulación Europea RGPD en cuanto a sus definiciones y principios de protección llegando a la conclusión que están estrechamente

relacionadas en estos aspectos. Así también, aprovechando trabajos ya realizados en el ámbito RGPD y mediante el estudio del PLOPDP se extrae una lista de tratamientos de datos personales, no exhaustiva, con el propósito de validar la herramienta desarrollada. Finalmente, se presentan de manera breve las herramientas y lenguajes de programación utilizados para la implementación de esta herramienta.

1.3.1. DEFINICIONES

A continuación, se presentan algunas definiciones importantes para el entendimiento de secciones posteriores. Las definiciones presentadas fueron tomadas del PLOPDP [4].

- **Dato personal:** *“Dato que identifica o hace identificable a una persona natural, directa o indirectamente, en el presente o futuro. Los datos inocuos, metadatos o fragmentos de datos que identifiquen o hagan identificable a un ser humano, forman parte de este concepto”.*
- **Tratamiento:** *“Cualquier operación o conjunto de operaciones realizadas sobre datos personales o conjuntos de datos personales, ya sea por procedimientos técnicos de carácter automatizado, parcialmente automatizado o no automatizado, tales como: la recogida, recopilación, obtención, registro, organización, estructuración, conservación, custodia, adaptación, modificación, eliminación, indexación, extracción, consulta, elaboración, utilización, posesión, aprovechamiento, comunicación por transmisión, transferencia, difusión, procesamiento, almacenamiento, distribución, cesión, o cualquier otra forma de habilitación de acceso, cotejo, interconexión, limitación, supresión, destrucción y, en general, cualquier uso de datos personales”*
- **Responsable del tratamiento de datos personales:** *“Persona natural o jurídica, pública o privada, que decide sobre la finalidad y el tratamiento de datos personales”*
- **Encargado del tratamiento de datos personales:** *“Persona que trate datos personales por nombre y a cuenta de un responsable de tratamiento de datos personales”*
- **Tercero:** *“Persona que no ostenta la calidad de responsable o encargado de tratamiento; titular, o, Autoridad de Protección de Datos Personales”*

1.3.2. PRIVACIDAD Y PROTECCIÓN DE DATOS PERSONALES

La privacidad es reconocida como un derecho humano fundamental [13]. Como tal ha sido estudiada durante muchos años desde una perspectiva interdisciplinar, incluyendo los ámbitos: social, legal y de ingeniería, con dos propósitos principales: 1) entender de mejor manera qué es la privacidad para los individuos y sociedades y 2) asegurarse de que los marcos legales dan soporte a la privacidad como un derecho.

Para abordar esta y las siguientes secciones es necesario mencionar tres definiciones básicas: privacidad, datos personales y protección de datos personales. Establecer una definición única de privacidad es una tarea extremadamente difícil como lo sostiene Solove [14]. Sin embargo, una definición útil para este caso de estudio es la siguiente: *“la privacidad es la habilidad que un individuo tiene para controlar el uso, retención, difusión o cualquier acción sobre sus datos personales”* [14]. Por otro lado, el PLODP define a los datos personales así: *“dato que identifica o hace identificable a una persona natural, directa o indirectamente, en el presente o en el futuro. Los datos inocuos, metadatos o fragmentos de datos que identifiquen o hagan identificable a un ser humano, forman parte de este concepto”* [4]. Finalmente, la protección de datos se define como cualquier medida organizacional o técnica para proteger el derecho a la privacidad.

Desde el punto de vista ingenieril, la privacidad puede ser abordada bajo tres paradigmas: 1) privacidad como confidencialidad, tal vez la más conocida y ampliamente desarrollada, su objetivo es el ocultamiento de información, 2) privacidad como control, cuyo objetivo es proporcionar al usuario final los medios para decidir sobre los datos personales que serán expuestos y 3) privacidad como transparencia, que busca informar al usuario sobre sus datos personales que han sido expuestos y quien ha tenido acceso o los ha procesado [5]. A continuación, se presentan de manera breve cada uno de estos paradigmas.

1. Privacidad como confidencialidad

La privacidad como confidencialidad busca el ocultamiento de información a un público más amplio del deseado. Es decir, asegurar el uso de servicios basados en Internet mientras se minimiza la cantidad de información expuesta. Esta información se subdivide en: información explícitamente intercambiada entre el usuario y el servicio e información implícitamente expuesta en los metadatos² asociados con el intercambio entre el usuario y el servicio (por ejemplo: la dirección IP del usuario o la frecuencia de uso del servicio).

² **Metadatos:** datos que describen otros datos.

Para reducir la cantidad de información expuesta explícitamente intercambiada entre el usuario y el servicio, existen dos grandes grupos de técnicas: 1) métodos que previenen el acceso no autorizado a datos personales mediante el uso de técnicas de criptografía, que aseguran que esta información no pueda ser inferida, y 2) métodos de control de la compartición de datos mediante la ofuscación de información, que buscan que la información expuesta esté limitada o que no se pueda relacionar a un individuo con ésta, algunos ejemplos de estos métodos son: anonimización³, generalización⁴, supresión⁵, adición ficticia⁶, perturbación⁷ entre otras.

Por otro lado, estas técnicas no son suficientes para asegurar la confidencialidad de los metadatos, en la mayoría de los casos. Existen tres tipos de metadatos que son comprobados de ser vulnerables de ataques de privacidad [5]:

- a. Metadatos de tráfico: asociados a la infraestructura de comunicación. Una técnica para proteger este tipo de datos es mediante el uso de redes de comunicación anónimas. Por ejemplo: Tor⁸ o redes privadas virtuales en general (VPNs, por sus siglas en inglés).
- b. Metadatos del dispositivo: son características propias del dispositivo (sistema operativo, versión del navegador utilizado, etc) del usuario que son enviados junto a las solicitudes a un servicio basado en Internet con el fin de optimizar la respuesta desde el servidor. No existe actualmente una técnica que ayude a mantener este tipo de datos confidenciales sin afectar la utilidad de los servicios [5].
- c. Metadatos de localización: la ubicación física de un usuario puede ser usada por servicios en línea para inferir información sensible. En muchos casos la

³ **Anonimización:** busca desacoplar la identidad de un individuo de sus datos. La idea es que al eliminar información que lo hace identificable de un conjunto de datos el individuo no pueda ser vinculado con estos.

⁴ **Generalización:** esta técnica consiste en reducir la exactitud de la información que se comparte con el objetivo de reducir la precisión con la que un atacante puede inferir esta información.

⁵ **Supresión:** esta técnica consiste en retirar información antes de exponerla. La lógica detrás de esta técnica es que mientras menor sea la cantidad de información expuesta será más difícil inferir la misma, en caso de una interferencia externa.

⁶ **Adición ficticia:** consiste en añadir información falsa a información real con el objetivo de ocultar los registros verdaderos.

⁷ **Perturbación:** esta técnica inyecta ruido en la información, de esta manera si la información es interceptada los valores reales no pueden ser inferidos directamente.

⁸ <https://www.torproject.org/>

localización es un quasi-identificador⁹. En este caso las técnicas de encriptación y de ofuscamiento de información pueden ser aplicadas.

2. Privacidad como control

Una noción más amplia de privacidad, considerada generalmente en las regulaciones, va más allá del ocultamiento de información personal y se expande a la capacidad del usuario para controlar qué información personal comparte, quiénes son los destinatarios, durante cuánto tiempo se comparten los datos, entre otros. Este paradigma tiene en cuenta que, en muchos casos, la compartición de información es necesaria para la provisión de un servicio.

Existen, principalmente, tres técnicas que soportan el paradigma de la privacidad como control.

a. Soporte para la configuración de ajustes de privacidad

Los ajustes de privacidad son los controles de un sistema de información que permiten a los usuarios configurar sus preferencias en cuanto a cómo sus datos personales deben ser revelados a otros usuarios, compartidos con terceros o procesados por el proveedor del servicio [5]. Sin embargo, en la mayoría de casos, la complejidad que presentan estos ajustes de privacidad hacen que para el usuario final sean poco prácticos, lo que resulta en una configuración incorrecta y por lo tanto en una compartición de datos no deseada [15].

Para contrarrestar estas dificultades, investigadores del área proponen técnicas para identificar grupos de usuarios que tienen características en común y en base a esto establecer de manera automática o semiautomática las configuraciones de privacidad más adecuadas [5].

b. Soporte para la negociación de políticas de privacidad

Estas técnicas se enfocan en la automatización de la comunicación de las preferencias del usuario con el servicio. Su objetivo es automatizar el proceso de configuración de los ajustes de privacidad, a través de la comunicación de sus preferencias de privacidad preconfiguradas.

⁹ **Quasi-identificador:** información que no es un identificador único por sí mismo, pero está suficientemente correlacionado a una entidad que al combinarse con otros quasi-identificadores pueden crear un identificador único.

Un ejemplo es el proyecto Plataforma para Preferencias de Privacidad (P3P) impulsado por el consorcio W3C (*World Wide Web Consortium*). Permite a los sitios web codificar sus políticas de privacidad en un formato predefinido, de manera que estas políticas puedan ser leídas de manera automática por el navegador. Entonces, pueden ser comparadas con las preferencias de privacidad preconfiguradas del usuario. Sin embargo, este proyecto en la actualidad no se encuentra ampliamente difundido y como se muestra en su sitio oficial¹⁰ el trabajo se encuentra suspendido.

c. Soporte para la interpretación de políticas de privacidad

Para que los usuarios configuren los ajustes de privacidad de acuerdo con sus expectativas de cómo sus datos deberían ser tratados, es necesario que comprendan las políticas de privacidad que describen estos ajustes. Sin embargo, las políticas de privacidad generalmente son extensas, verbosas y usan términos legales que evolucionan en el tiempo, por lo tanto los usuarios no pueden interpretarlas de manera fácil [5].

Para mejorar el entendimiento de las políticas de privacidad existen dos técnicas actualmente: 1) la anotación manual con soporte de expertos en el área y 2) la anotación mediante herramientas de aprendizaje automático.

Es precisamente en la técnica de anotación manual en la que este trabajo de titulación se sustenta. A través de una herramienta web se podrán realizar anotaciones de los tratamientos de datos personales expuestos en el PLOPDP. Los resultados de estas anotaciones serán expuestos en un sitio web que servirá para visualizar de una manera estructurada y legible los tratamientos de datos expresados en estos documentos. Al mismo tiempo, la herramienta generada será usada para la creación de un corpus de políticas de privacidad anotadas que a su vez es un insumo para la generación de modelos de aprendizaje automático.

3. Privacidad como transparencia

Finalmente, la privacidad puede ser abordada bajo el paradigma de la transparencia. Los mecanismos de transparencia analizan la actividad que los usuarios realizan en línea con dos fines: 1) retroalimentar al usuario acerca de sus acciones y las implicaciones que estas conllevan y 2) realizar auditorías para asegurar que los

¹⁰ <https://www.w3.org/P3P/>

ajustes de privacidad del usuario no fueron violados. Se asume que la actividad de los usuarios es analizada por una autoridad en la que se confía.

1.3.3. PROYECTO DE LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES

El 19 de septiembre de 2019 el gobierno ecuatoriano a través del MINTEL presentó a la Asamblea Nacional el proyecto de Ley Orgánica de Protección de Datos Personales [4]. Esto sucedió días después, exactamente el 16 de septiembre, que se diera a conocer una filtración masiva de datos de ciudadanos ecuatorianos donde los datos personales de más de 20 millones de personas quedaron expuestos desde los servidores de la compañía Novaestrat, estos incluían datos de personas fallecidas y de menores de edad [16].

La última reunión para tratar temas referentes al PLOPDP, de acuerdo con el sitio oficial de la asamblea nacional¹¹, se realizó el 23 de septiembre de 2020. En esta mesa de trabajo se habló de la importancia del PLOPDP, se realizaron sugerencias y observaciones acerca de los artículos que conforman la ley, de las entidades de control que deberían regular este aspecto. Sin embargo, luego de esta mesa de trabajo no existe evidencia de avances para la aprobación de la ley, por lo que sigue siendo un proyecto. Este trabajo de titulación se basa en la versión del documento que fue presentado a la asamblea el día 19 de septiembre de 2019 y que a la fecha es la versión con la que trabaja el pleno.

Este marco legal surge como un mecanismo jurídico para proteger el derecho a la privacidad, definiendo un conjunto de principios, derechos y obligaciones y para los diferentes actores del tratamiento de datos personales [4].

El ámbito de aplicación del PLOPDP se extiende para cualquier tratamiento de datos personales ya sean totalmente automatizados, parcialmente automatizados o no automatizados y a toda modalidad de uso posterior ya sea que el responsable o encargado del tratamiento pertenezca al sector público o privado. El ámbito de exclusión del PLOPDP se aplica para [4]:

1. Tratamiento de datos personales utilizados en actividades familiares o domésticas.
2. Datos anónimos.
3. Datos que identifican o hacen identificable a personas jurídicas.

En cuanto a la aplicación territorial, la ley tiene una aplicación [4]:

¹¹ <https://www.asambleanacional.gob.ec/>

1. Para todo tratamiento de datos personales que se realice en cualquier parte del territorio nacional.
2. Cuando el responsable o encargado del tratamiento se encuentre domiciliado en cualquier parte del territorio nacional.
3. Cuando el responsable o encargado del tratamiento no domiciliado en el territorio nacional oferte a ecuatorianos bienes o servicios, o realice actividades relativas al control de su comportamiento.
4. Cuando al responsable o encargado del tratamiento no domiciliado en el territorio nacional le resulte aplicable la legislación nacional en virtud de la celebración de un contrato o del derecho internacional público.
5. Cuando el responsable o encargado del tratamiento no esté domiciliado en el territorio nacional y utilice o recurra a medios automatizados, parcialmente automatizados o no automatizados, situados en el Ecuador para tratar datos personales, salvo que dichos medios se utilicen solamente con fines de tránsito.

De este proyecto de ley cabe destacar la clara influencia de la existente normativa europea (RGPD), así también, que busca la protección del ciudadano y no de las empresas al reconocer en sus principios básicos, en casos de duda, la aplicación favorable al titular.

1.3.3.1. Datos personales

Actualmente en el Ecuador la protección de datos personales está regulada de manera superficial, imprecisa e insuficiente. Algunas normas jurídicas que ya mencionan a la protección de datos se presentan a continuación:

a) Constitución de la república del Ecuador

El artículo 66 numeral 19 habla de la protección de datos personales de la siguiente manera: *“el derecho a la protección de datos de carácter personal, que incluye el acceso y la decisión sobre información y datos de este carácter, así como su correspondiente protección. La recolección, archivo, procesamiento, distribución o difusión de estos datos o información requerirán la autorización del titular o el mandato de la ley”* [17].

b) Ley de comercio electrónico, firmas electrónicas y mensajes de datos

En el contexto ecuatoriano, esta ley en el artículo 66 es la única que provee una definición sobre datos personales: *“Datos personales: son aquellos datos o*

información de carácter personal o íntimo, que son materia de protección en virtud de esta Ley". [18]

c) Ley del sistema nacional del registro de datos públicos

A pesar de no contar con una definición de datos personales, en su artículo 13 esta ley hace referencia a los datos públicos: *"De los registros de datos públicos.- Son registros de datos públicos: el Registro Civil, de la Propiedad, Mercantil, Societario, Vehicular, de naves y aeronaves, patentes, de propiedad intelectual y los que en la actualidad o en el futuro determine la Dirección Nacional de Registro de Datos Públicos, en el marco de lo dispuesto por la Constitución de la República y las leyes vigentes"* [19]

Esta definición es importante ya que establece como jurisdicción de la Dirección Nacional de Registro de Datos Públicos (DINARDAP) el decidir los datos personales que son o serán considerados como datos públicos.

d) Código Orgánico Integral Penal (COIP)

El COIP tipifica el delito de violación a la intimidad en su artículo 178: *"La persona que, sin contar con el consentimiento o la autorización legal, acceda, intercepte, examine, retenga, grabe, reproduzca, difunda o publique datos personales, mensajes de datos, voz, audio y vídeo, objetos postales, información contenida en soportes informáticos, comunicaciones privadas o reservadas de otra persona por cualquier medio, será sancionada con pena privativa de libertad de uno a tres años"* [20]

El problema que genera este artículo es que no se tiene una definición clara de lo que son los datos personales.

Las definiciones antes presentadas establecen el derecho a la protección de datos personales, sin embargo, no existe una definición suficientemente clara de lo que son los datos personales.

Por otro lado, el PLOPD define a los datos personales así: *"dato que identifica o hace identificable a una persona natural, directa o indirectamente, en el presente o en el futuro. Los datos inocuos, metadatos o fragmentos de datos que identifiquen o hagan identificable a un ser humano, forman parte de este concepto"* [4].

Los puntos relevantes de esta definición son los siguientes:

1. Sujeto de protección: solo aplica para personas naturales y no para personas jurídicas.
2. Dato que identifica: cualquier tipo de dato que permite identificar a una persona de manera directa, por ejemplo: el número de cédula o pasaporte.
3. Dato que hace identificable: cualquier tipo de dato que no identifica de manera directa a una persona, pero que en conjunto con otros datos puede llegar a identificarla, por ejemplo: dirección postal, localización.
4. Metadatos: esto incluye identificadores que se usan para rastrear a una persona y construir perfiles. Por ejemplo, identificadores web o identificadores de aplicaciones móviles que sirven para construir perfiles de comportamiento.

Esta definición permite tener una idea más clara de lo que son los datos personales y está enmarcada en un contexto digital, donde priman los servicios que se ofrecen en Internet. Así mismo el artículo 5 del PLOPDP proporciona definiciones complementarias: dato biométrico, dato genético, dato personal crediticio, dato personal registrable y dato sensible, que aclaran definiciones de normas jurídicas anteriores.

1.3.3.2. Políticas de privacidad

Las políticas de privacidad, llamadas políticas de tratamiento de datos en el PLOPDP, son el principal medio que se ha adoptado para informar al usuario la manera en que un servicio hace uso de sus datos personales. De acuerdo con el PLOPDP *“el responsable del tratamiento de datos debe tomar las medidas apropiadas para proporcionar a sus usuarios cualquier información en relación con el procesamiento de datos, en una forma concisa, transparente, inteligible y de fácil acceso, utilizando un formato claro e idioma sencillo, la información se proporcionará por escrito o por otros medios, esta información deberá ser comunicada el momento mismo de la recogida del dato personal”*[4] .

Estudios en el área de la privacidad y protección de datos personales demuestran que los usuarios son cada vez más conscientes de la importancia de su privacidad al hacer uso de servicios en línea. Estos estudios demuestran que los usuarios en la mayoría de las ocasiones terminan ignorando las políticas de privacidad y una de las principales razones es la limitada legibilidad, también concluyen que en promedio el nivel de lectura requerido para leer y entender una política de privacidad es de un nivel de educación universitario o superior. [6]

De acuerdo con el Instituto Nacional de Estadísticas y Censos del Ecuador (INEC) solamente el 14,5% de la población ecuatoriana tiene un título de tercer nivel [21]. Esto

podría convertirse en una barrera para la interpretación de los tratamientos de datos que una empresa ejecuta sobre los datos personales de los usuarios para la gran mayoría de la población ecuatoriana, por lo que este trabajo puede ser relevante para contribuir a disminuir esta barrera.

1.3.3.3. Principios de protección

El PLOPDP define una serie de principios, en total 14, que deben prevalecer cuando se realice cualquier tratamiento de datos, a continuación, se presenta una descripción breve de cada uno de estos principios:

1. **Juridicidad:** los datos personales en ningún caso podrán ser tratados para fines ilícitos o desleales.
2. **Legitimidad:** para que un tratamiento de datos sea considerado legítimo debe cumplir con alguna de estas condiciones:
 - a. Exista obligación legal aplicable al responsable del tratamiento
 - b. Por orden judicial, resolución o mandato motivado de autoridad pública competente.
 - c. Para el ejercicio de las competencias y facultades establecidas en la Constitución, la Ley, instrumentos internacionales ratificados por el Ecuador y demás normativa aplicable a favor de las entidades pertenecientes al sector público, sus delegatarios y organizaciones de Derecho Internacional Público.
 - d. Para el cumplimiento de obligaciones contractuales perseguidas por el responsable del tratamiento de datos personales, encargado del tratamiento de datos personales o por un tercero legalmente habilitado.
 - e. Para la ejecución de medidas precontractuales a petición del titular; excepto cuando prevalezcan los intereses o los derechos y libertades de niñas, niños y adolescentes como titulares.
 - f. Por consentimiento del titular para el tratamiento de sus datos personales para una o varias finalidades específicas.
 - g. Para proteger intereses vitales, del interesado o de otra persona natural, como por ejemplo su vida, salud o integridad.

3. **Finalidad:** las finalidades del tratamiento deberán ser determinadas, explícitas y legítimas y no podrán tratarse datos personales con fines distintos para los cuales fueron recopilados.
4. **Pertinencia y minimización:** los datos personales deben ser pertinentes y limitados a lo mínimo necesario para su finalidad.
5. **Proporcionalidad del tratamiento:** el tratamiento debe ser adecuado, necesario, oportuno, relevante y no excesivo en relación con las finalidades para las cuales fueron recogidos o a la naturaleza de las categorías especiales de datos.
6. **Consentimiento:** se podrán tratar y comunicar datos personales cuando se cuente con la manifestación de la voluntad del titular de hacerlo.

Para que el consentimiento sea considerado válido, la manifestación de la voluntad debe cumplir con ciertas condiciones, tiene que ser: libre, específica, informada, inequívoca, previa y expresa.

Es importante mencionar que este consentimiento puede revocarse en cualquier momento, para lo cual el responsable del tratamiento debe proveer un mecanismo igual de sencillo que el que fue llevado a cabo para la recolección del consentimiento.

7. **Confidencialidad:** el tratamiento de datos personales debe darse en un contexto de sigilo y secreto. Este principio implica también la facultad que tiene el titular de controlar la forma en que se tratan sus datos, incluyendo la transferencia o comunicación.
8. **Calidad:** los datos personales que son objeto de un tratamiento deben ser exactos, precisos, íntegros, comprobables, claros y de ser necesarios actualizados, sin que esto altere su veracidad.
9. **Conservación:** los datos personales pueden ser conservados en los siguientes casos:
 - a. Durante el tiempo consentido, determinado en el ordenamiento jurídico o establecido en orden judicial, resolución o mandato de autoridad competente.
 - b. Hasta cuando cumplan con la finalidad para la cual fueron recogidos o tratados.

Una vez cumplido cualquiera de estos casos los datos personales deberán ser suprimidos o ser sometidos a procesos de anonimización.

10. **Seguridad de datos personales:** los responsables y encargados del tratamiento de datos personales deberán implementar todas las medidas de seguridad adecuadas y necesarias para proteger estos datos de cualquier riesgo o amenaza que puedan sufrir.

11. **Responsabilidad proactiva y demostrada:** el responsable del tratamiento de datos personales debe estar en la capacidad de demostrar la implementación de mecanismos para la protección de estos datos. El responsable del tratamiento está en la obligación de rendir cuentas sobre los tratamientos al titular y a la Autoridad de Protección de Datos Personales.

Así mismo, el responsable del tratamiento debe evaluar sus procesos en todo momento y adoptar el principio de responsabilidad de forma continua y permanente, para mejorar el nivel de protección de los datos personales.

12. **Aplicación favorable al titular:** en caso de duda sobre el alcance de las disposiciones del ordenamiento jurídico o contractuales, en cuanto a la protección de datos personales, los funcionarios judiciales y administrativos interpretarán y aplicarán en el sentido más favorable al titular de estos datos.

13. **Independencia de control:** para que el ejercicio de la protección de datos personales sea efectivo, el estado ejercerá un control independiente, imparcial y autónomo, así como la regulación y sanción.

14. **Normativa especializada:** los datos personales que se encuentren regulados por normativas especializadas estarán sujetos a los principios establecidos en sus propias normas.

1.3.4. PLOPDP VS. RGPD

Al comparar el PLOPDP con el RGPD es fácil notar la influencia que tiene la normativa europea sobre el proyecto de ley ecuatoriano. Las definiciones y principios de protección al usuario son similares. Como un ejemplo podemos analizar la definición de **tratamiento de datos personales:**

El RGPD define al tratamiento de datos personales como: *“cualquier operación o conjunto de operaciones realizadas sobre datos personales o conjuntos de datos personales, ya sea por procedimientos automatizados o no, como la recogida, registro, organización,*

estructuración, conservación, adaptación o modificación, extracción, consulta, utilización, comunicación por transmisión, difusión o cualquier otra forma de habilitación de acceso, cotejo o interconexión, limitación, supresión o destrucción” [3].

Por otro lado, el PLOPDP define a un tratamiento de datos así: *“cualquier operación o conjunto de operaciones realizadas sobre datos personales o conjuntos de datos personales, ya sea por procedimientos técnicos de carácter automatizado, parcialmente automatizado o no automatizado, tales como: la recogida, recopilación, obtención, registro, organización, estructuración, conservación, custodia, adaptación, modificación, eliminación, indexación, extracción, consulta, elaboración, utilización, posesión, aprovechamiento, comunicación por transmisión, transferencia, difusión, procesamiento, almacenamiento, distribución, cesión, o cualquier otra forma de habilitación de acceso, cotejo, interconexión, limitación, supresión, destrucción y, en general, cualquier uso de datos personales” [4].*

Como se puede observar las dos definiciones son muy similares, la única diferencia es que la definición propuesta por el PLOPDP ejemplifica y amplía las acciones que son consideradas como un tratamiento de datos personales.

En la tabla 1.2 se presenta una lista de los conceptos y si está o no definido en cada normativa:

Tabla 1.1 Contrastación de definiciones PLOPDP vs RGPD

Concepto	PLOPDP	RGPD
Anonimización	Si	Si, pero está definido como seudonimización.
Autoridad de control	No y no existe una definición comparable	Si
Autoridad de control interesada	No y no existe una definición comparable.	Si
Base de datos	Si	Si, pero está definido como fichero.
Consentimiento	Si	Si
Dato biométrico	Si	Si
Dato genético	Si	Si
Dato personal	Si	Si
Dato personal crediticio	Si	No y no existe una definición comparable.
Dato personal registrable	Si	No y no existe una definición comparable.
Datos relativos a la salud	No y no existe una definición comparable.	Si
Dato sensible	Si	No y no existe una definición comparable.
Destinatario	Si	Si
Disociación de datos	Si	No, pero la seudonimización engloba este concepto.

Elaboración de perfiles	Si	Si
Empresa	No y no existe una definición comparable.	Si
Encargado del tratamiento de datos personales	Si	Si
Establecimiento principal	No y no existe una definición comparable	Si
Estado de la técnica	Si	No y no existe una definición comparable.
Fichero	Si, pero está definido como base de datos.	Si
Filtración	Si	No, pero la violación a la seguridad engloba este concepto.
Fuentes accesibles al público	Si	No y no existe una definición comparable.
Grupo empresarial	No y no existe una definición comparable.	Si
Limitación del tratamiento	No	Si
Normas corporativas vinculantes	No y no existe una definición comparable.	Si
Objeción pertinente y motivada	No y no existe una definición comparable.	Si
Organización internacional	No y no existe una definición comparable.	Si
Política de tratamiento de datos personales	Si	No y no existe una definición comparable.
Representante	No y no existe una definición comparable.	Si
Responsable del tratamiento de datos personales	Si	Si
Sellos de datos personales	Si	No y no existe una definición comparable.
Servicio de la sociedad de la información.	No y no existe una definición comparable	Si
Seudonimización	Si, pero está definido como anonimización.	Si
Tercero	Si	Si
Titular	Si	No y no existe una definición comparable.
Transferencia o comunicación	Si	No, pero existe un concepto que es una derivación: tratamiento transfronterizo.
Tratamiento transfronterizo	No, pero la transferencia engloba este concepto.	Si
Violación de la seguridad de los datos personales	Si, pero está definido como una vulneración de la seguridad.	Si
Vulneración de la seguridad de los datos personales	Si	Si, pero está definida como violación a la seguridad.

Existen conceptos que no son definidos por una u otra normativa, fueron descritos como “no y no existe una definición comparable”, y es que la mayoría de estos conceptos no se abordan en cualquiera de los dos marcos legales porque son únicos para el contexto en que estos fueron creados. Un ejemplo en el RGPD es la definición de **establecimiento**

principal que aborda cómo determinar el establecimiento principal de un responsable del tratamiento cuando este cuenta con establecimientos en más de un país que conforma la UE. Esta definición no aplica para el contexto ecuatoriano y es por lo que no se puede encontrar una definición similar en el documento.

Un caso interesante es que el RGPD no incluye una definición de política de tratamiento de datos personales, esto a pesar de que son el principal medio por el un usuario es informado sobre los tratamientos que se ejecutan sobre sus datos. Sin embargo, a pesar de no existir una definición como tal, los artículos 12, 13 y 14 tratan ampliamente este tema y dan una serie de guías y requerimientos que una política de privacidad debería tener para estar en conformidad con el GDPR.

Por otro lado, también se puede hacer un análisis comparativo de los principios de protección que estas normativas exponen. El RGPD presenta los principios de protección en su capítulo III, en los artículos 5, 6, 7, 8, 9, 10 y 11. Así mismo, el PLODPA presenta los principios de protección en su capítulo II en los artículos del 8 al 22. La tabla 1.3 muestra una comparación de los principios expuestos en cada una de estas normativas.

Tabla 1.2 Comparativa de principios de protección PLODPA vs. RGPD

Principio de protección	PLODPA	RGPD
Juridicidad, lealtad y transparencia	Si	Si, pero está definido como licitud, lealtad y transparencia.
Legitimidad	Si	Si, pero está definido como la licitud del tratamiento.
Finalidad	Si	Si, pero está definido como limitación de la finalidad.
Pertinencia y minimización de datos personales	Si	Si, pero está definido como minimización de datos
Proporcionalidad del tratamiento	Si	No, pero está abarcado por el concepto de limitación de la finalidad.
Consentimiento	Si	Si, pero está definido como la licitud del tratamiento.
Confidencialidad	Si	Si, pero está definido como integridad y confidencialidad.
Calidad	Si	Si, pero está definido como exactitud.
Conservación	Si	Si, pero está definido como limitación del plazo de conservación.
Seguridad de datos personales	Si	No, pero es mencionado en el principio de integridad y confidencialidad.
Responsabilidad proactiva	Si	Si
Aplicación favorable al titular	Si	No y no existe una definición comparable.
Independencia de control	Si	No y no existe una definición comparable.

Normativa especializada	Si	No y no existe una definición comparable.
Condiciones aplicables al consentimiento del niño en relación con los servicios de la sociedad de la información	Si, pero es tratado en el capítulo IV artículo 39 sobre el consentimiento relativo de niños, niñas y adolescentes.	Si
Tratamiento de categorías especiales de datos personales	Si, pero este concepto es abordado en el capítulo IV	Si
Tratamiento de datos personales relativos a condenas e infracciones penales	No, pero el concepto de legitimidad abarca esta definición.	Si
Tratamiento que no requiere identificación	No y no existe una definición comparable	Si

Como se puede observar la normativa ecuatoriana es más explícita en cuanto a la explicación de los principios de protección ya que presenta cada principio en un artículo independiente, sin embargo, esto no hace una diferencia significativa ya que de una u otra manera los mismos principios son tratados en ambas normativas, esto se muestra en la Tabla 1.1.

Unos de los principios que tiene que resaltarse es el de responsabilidad proactiva, que obliga al responsable del tratamiento de datos personales a estar pendiente en todo momento de mantener las mejores prácticas para mantener seguros los datos personales de sus usuarios y así mismo a estar en la capacidad de rendir cuentas al usuario y a la Autoridad de protección de Datos Personales.

1.3.5. CATEGORÍAS DE TRATAMIENTO DE DATOS PERSONALES

El PLOPDP define requerimientos legales que necesitan ser cumplidos por el responsable del tratamiento de datos. Las políticas de privacidad precisan los tratamientos de datos personales basadas en estos requerimientos y generalmente estos tratamientos tienen la siguiente estructura:

1. Tratamiento
 - 1.1. Atributo
 - 1.1.1. Valor

Estos tratamientos no están explícitamente expresados en el PLOPDP, por lo que tienen que extraerse de los distintos artículos que componen esta ley. El proceso de elicitación de requerimientos a partir de normas legales o estándares es sumamente complejo y generalmente requiere la colaboración de un equipo multidisciplinar que incluya, al menos, las perspectiva legal y técnica.

Generalmente las categorías de tratamientos de datos están expresadas por un sustantivo derivado de un verbo, por ejemplo: recolección. Para fines prácticos una categoría de tratamiento de datos será llamada tratamiento en la herramienta de anotación.

Los atributos son subcategorías de un tratamiento de datos y ayudan a describirlo de manera más detallada. Este término fue adoptado en el presente trabajo con la finalidad de hacer más granular la anotación de tratamiento de datos. Atributo no es un término definido en el PLOPDP.

Existe un tercer nivel que son los valores de un atributo. Son descripciones aún más refinadas de un atributo. Sin embargo, por la complejidad que representa la definición de los valores de un atributo, todos los atributos de la herramienta tendrán el valor "Genérico".

A continuación, se presentan las principales categorías de tratamiento de datos y atributos identificados en el PLOPDP. Vale la pena mencionar que este conjunto de tratamientos y atributos no es exhaustivo, como se ha mencionado, requeriría un trabajo multidisciplinar (legal principalmente) y esto escapa del alcance de este proyecto.

La **recolección** de datos es mencionada en el artículo 23 y algunos de sus atributos pueden ser derivados de este mismo artículo como: el **tipo de dato** recolectado, la **finalidad** o **propósito** de la recolección, el **origen de los datos** y la **elección del usuario** en cuanto a esta recolección.

Del artículo 17 se puede obtener el tratamiento de datos: **conservación**, este mismo artículo detalla sus atributos que son: **período de conservación** y **propósito** de conservación. Esta categoría de tratamiento de datos también se menciona en el artículo 23 numeral 4, sin embargo, se tomó el artículo 17 como referencia ya que presenta mayores detalles.

Del artículo 23 también puede extraerse el tratamiento de datos: **transferencia**, este tratamiento es el más ampliamente tratado en el PLOPDP. Sus atributos son: **transferencia a terceros**, **transferencia internacional** y **propósito** de la transferencia. La "*transferencia, comunicación y acceso a terceros*" se encuentra ampliamente explicado en el capítulo V, en un total de 4 artículos del PLOPDP, mientras que la "*transferencia o comunicación internacional*" está definida en el capítulo VIII del PLOPDP, en un total de 7 artículos. La principal diferencia entre la transferencia a terceros y la transferencia internacional es que transferencia internacional de datos se refiere a que los datos son enviados fuera del territorio nacional, pero el responsable del tratamiento de datos sigue siendo el mismo que en un inicio, mientras la transferencia a terceros la ubicación

geográfica del destinatario puede estar o no en el territorio nacional y éste se convierte en el encargado del tratamiento de datos personales, para este tipo de transferencia específico es necesario aclarar que se necesita el consentimiento por parte del titular.

Finalmente, el último tratamiento identificado en el artículo 23 es: **control del titular** y los atributos para este tratamiento son: **acceso, edición y eliminación**. Estos atributos se encuentran definidos en el capítulo IV del PLOPDP en los artículos 24, 25 y 26 respectivamente.

Las categorías de tratamientos de datos personales con sus respectivos atributos y valores obtenidos del PLOPDP se presentan en la Tabla 1.3:

Tabla 1.3 Resumen de tratamientos de datos

Tratamiento de datos	Atributo	Valor
Recolección	Tipo de dato	Genérico
	Propósito	Genérico
	Origen de los datos	Genérico
	Elección del usuario	Genérico
Conservación	Periodo de conservación	Genérico
	Propósito	Genérico
Transferencia	Transferencia internacional	Genérico
	Transferencia a terceros	Genérico
	Propósito	Genérico
Control del titular	Acceso	Genérico
	Edición	Genérico
	Eliminación	Genérico

Los valores identificados en la tabla anterior serán los que se usen en un futuro para la validación de la herramienta, a través de la anotación de cinco políticas de privacidad. La herramienta de anotación fue diseñada para soportar tratamientos de datos que contengan los tres niveles antes descritos: tratamiento, atributo y valor.

1.3.6. METODOLOGÍA SCRUM

Scrum es un marco de trabajo diseñado para la gestión del desarrollo, mantenimiento o mejora de productos complejos y puede ser aplicado en cualquier ámbito [22].

Scrum define una serie de artefactos para proveer transparencia:

- *Product backlog*: es una lista ordenada de historias de usuario. Es la fuente de requerimientos y cambios del producto.

- *Sprint backlog*: es el conjunto de historias de usuario seleccionadas del *product backlog* para ser desarrolladas durante el *sprint*.

Scrum define un grupo de trabajo llamado equipo Scrum y está compuesto por [22]:

- *Product owner*: es el responsable de priorizar y ordenar las historias de usuario en el *product backlog*.
- Equipo de desarrollo: son los responsables de realizar las tareas del *product backlog*.
- *Scrum master*: es el responsable de que el proceso de Scrum sea entendido y aplicado de manera correcta por el equipo.

Así mismo, *Scrum* proporciona un conjunto de eventos para crear regularidad en el desarrollo del producto, en este trabajo se utilizaron los siguientes eventos [22]:

- *Sprint*: es la unidad básica de trabajo, consiste en un periodo de tiempo en el cual se crea un incremento de producto terminado.
- Reunión de planificación de un *sprint*: este evento sirve para planificar el trabajo que se llevará a cabo durante el *sprint*, esta reunión se realiza con el equipo Scrum completo.
- Revisión del *sprint*: en la revisión del *sprint* participarán el equipo de desarrollo y *product owner*, y tiene como objetivo revisar las historias de usuario desarrolladas durante el *sprint*.
- Retrospectiva del *sprint*: es una oportunidad para que el Equipo Scrum se inspeccione y cree un plan de mejoras para el siguiente *sprint*.

1.3.7. HERRAMIENTAS Y LENGUAJES DE PROGRAMACIÓN UTILIZADOS

En esta sección se presentarán de manera breve las herramientas, *frameworks* y lenguajes de programación utilizados para la implementación de la herramienta.

El desarrollo de la herramienta está basado en la arquitectura Modelo Vista Controlador (MVC). MVC ayuda a escribir código mejor organizado y por lo tanto más mantenible [23].

Los componentes de esta arquitectura se definen así:

- Modelo: define los datos que maneja el sistema, su lógica de negocio y los mecanismos de persistencia.

- Vista: es la interfaz que se presenta al usuario y la manera en que este interactúa con la aplicación.
- Controlador: actúa como intermediario entre la vista y el modelo, y gestiona el flujo de datos entre ellos.

A continuación, se presentan las herramientas usadas siguiendo la estructura MVC.

1.3.7.1. Modelo

En esta sección se presentan las herramientas y lenguajes usados para la implementación y gestión de la base de datos de la herramienta.

1.3.7.1.1. MySQL

MySQL¹² es el sistema más popular de administración de bases de datos de código abierto y es desarrollado, distribuido y soportado por Oracle. Las bases de datos implementadas con MySQL son relacionales, en consonancia con el diseño de la base de datos relacional diseñada en este trabajo. Por otro lado, SQL (Structured Query Language) es el lenguaje estandarizado mayormente usado para el acceso a las bases de datos y es usado en MySQL. Sin embargo, en este trabajo se utilizó Flask-SQLAlchemy que abstrae el uso de lenguaje SQL. Las operaciones con las bases de datos se realizan en un lenguaje definido por el ORM¹³.

1.3.7.1.2. Flask-SQLAlchemy

SQLAlchemy es el ORM para Python y proporciona todas las funcionalidades de SQL para ser ejecutadas directamente desde código de Python, de esta manera no existe la necesidad de operaciones adicionales con la base de datos. SQLAlchemy presenta un método para asociar clases de Python con tablas de la base de datos, e instancias de esas clases (objetos) con filas de sus tablas correspondientes.

1.3.7.2. Vista

En esta sección se explica de manera breve el lenguaje de programación (TypeScript) y el *framework* (Angular) usados en la implementación de la herramienta.

1.3.7.2.1. TypeScript

TypeScript es un superconjunto tipado de JavaScript que se compila a JavaScript puro. Permite realizar chequeo estático de tipos y refactorización al momento de escribir código.

¹² <https://www.mysql.com/>

¹³ **ORM**: El mapeo objeto-relacional es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia

Por lo tanto, con TypeScript es posible detectar errores en tiempo de compilación en lugar de en tiempo de ejecución.

De acuerdo con [24] el 15% de los errores producidos en JavaScript pueden ser detectados por TypeScript antes de que el código sea puesto en producción. Un sistema de tipos estático, como el que ofrece TypeScript, ofrece los siguientes beneficios:

- Detección de errores antes de ejecución.
- Aumento de eficiencia en tiempo de ejecución.
- Mejora la comprensión del código.
- Optimización al momento de compilar.

1.3.7.2.2. Angular

Angular es un *framework* de *frontend* para construir aplicaciones de una sola página (SPA, *Single Page Application*) haciendo uso de *HTML*, *CSS* y *TypeScript*. La arquitectura de una aplicación de Angular se basa en tres conceptos fundamentales: módulos, componentes y servicios. Los módulos, componentes y servicios son clases que usan el patrón de diseño Decorador. Un decorador en Angular permite marcar cada una de estas clases con su tipo correspondiente y provee de metadata que Angular interpreta. En el ANEXO C se encuentra una explicación más detallada de Angular.

1.3.7.3. Controlador

En esta sección se presentará de manera breve el lenguaje de programación utilizado para el desarrollo del *backend* de la aplicación (Python), así como el *framework* con el que se construyó los servicios RESTful (flask).

1.3.7.3.1. Python

Python es un lenguaje de programación interpretado, orientado a objetos, de alto nivel y con un tipado dinámico. Sus estructuras de datos de alto nivel en conjunto con el tipado y enlazado dinámico¹⁴ hacen de Python un lenguaje atractivo para el Desarrollo Rápido de Aplicaciones (RAD, por sus siglas en inglés), como lenguaje de *scripting* o para unir componentes existentes. Python tiene una sintaxis sencilla y fácil de aprender que enfatiza la legibilidad del código de esta manera reduce los costos de mantenimiento de las

¹⁴ **Enlazado dinámico:** también conocido como enlazado tardío, es un mecanismo en el cual un programa espera a tiempo de ejecución para enlazar efectivamente el nombre de un método o función a una subrutina de código.

aplicaciones desarrolladas con él. El intérprete de Python y la librería estándar están disponibles en formato fuente o binario y puede ser usado y distribuido sin cargo [25].

1.3.7.3.2. Flask

Flask es un *microframework* web WSGI¹⁵. Está diseñado para que iniciar una aplicación sea rápido y sencillo, con la capacidad para escalar hacia aplicaciones más complejas. Flask empezó como un encapsulamiento de las librerías Werkzeug¹⁶ y Jinja¹⁷ y hoy en día es uno de los *frameworks* para aplicaciones web más famosos para Python [26].

1.4. RELACIÓN CON TRABAJOS SIMILARES EN EL ÁREA

Como se expuso anteriormente existen trabajos que se encargan de la anotación de tratamientos de datos personales en políticas de privacidad. Los trabajos [7][8] permiten realizar anotaciones manuales de tratamientos de datos en políticas de privacidad en inglés y luego estas anotaciones son presentadas de manera estructurada al usuario final. Mientras que los trabajos [9][10][11][12] son capaces de realizar anotaciones de tratamientos de datos de manera automática, al hacer uso de sistemas de inteligencia artificial. Este trabajo se diferencia de los mencionados anteriormente en: 1) se busca realizar anotación manual de tratamientos de datos sobre políticas de privacidad en español y 2) cada uno de los trabajos anteriores fueron realizados bajo sus propias jurisdicciones. Este trabajo será desarrollado tomando como base el proyecto de Ley Orgánica de Protección de Datos Personales del Ecuador

¹⁵ **WSGI**: Web Server Gateway Interface, es una especificación que describe como un servidor web se comunica con las aplicaciones web y cómo las aplicaciones web pueden ser unidas para procesar una petición.

¹⁶ **Werkzeug**: es una librería completa para crear aplicaciones WSGI.

¹⁷ **Jinja**: es un motor de plantillas completo, creado para Python.

2. METODOLOGÍA

En este capítulo se presentan las fases de desarrollo de la herramienta, que será dividida en tres etapas: análisis, diseño e implementación.

2.1. METODOLOGÍA DE DESARROLLO

El desarrollo de esta herramienta está basado en la metodología de desarrollo ágil Scrum. En esta sección se describe como Scrum fue adaptado para el desarrollo de la herramienta:

- El *sprint backlog* contendrá todas las historias de usuario necesarias para crear un módulo de la herramienta. Cada historia de usuario del *sprint backlog* será manejada mediante la herramienta web Trello¹⁸ y pasará por las columnas “*Sprint #* (número de *sprint* correspondiente) *backlog*”, “Haciendo”, “Por revisar” y “Hecho” de acuerdo con la etapa en la que se encuentre.
- *Product owner*: en este proyecto esta persona es quien decide que una historia de usuario cumple con los requerimientos establecidos antes de su desarrollo.
- Equipo de desarrollo: en este trabajo el equipo de desarrollo está conformado por un miembro y será el encargado de estimar el esfuerzo necesario para desarrollar cada historia de usuario. El esfuerzo requerido para cada historia de usuario se cuantificó en base a una escala que corresponde a la serie de Fibonacci¹⁹. Para Mike Cohn estimar el esfuerzo con una secuencia no lineal es la mejor aproximación ya que refleja la incertidumbre asociada con unidades de trabajo más grandes [27]. Por lo tanto, el esfuerzo requerido para completar una historia de usuario podrá tener uno de los siguientes valores: 1,2,3,5 u 8, siendo 8 el esfuerzo requerido más alto.
- *Scrum Master*: En este trabajo el *Scrum master* tendrá la responsabilidad de colaborar con el equipo de desarrollo y el *product owner* para la planificación y desarrollo de la herramienta.
- *Sprint*: En este trabajo se decidió ajustar el periodo de desarrollo de un *sprint* de acuerdo con los esfuerzos estimados para las historias de usuario. Los *sprints* que presenten historias de usuario estimadas con 8 puntos serán los de mayor duración.

¹⁸ **Trello**: herramienta en línea para el desarrollo de proyectos basados en metodologías ágiles.

¹⁹ **Serie de Fibonacci**: es una secuencia ordenada de números que partiendo de los números 0 y 1 se puede calcular con la función $f_n = f_{n-1} + f_{n-2}$

- Revisión/Retrospectiva del *sprint*: para el desarrollo de este trabajo se decidió unificar los eventos de revisión y retrospectiva. Este nuevo evento denominado revisión/retrospectiva del *sprint* participarán el equipo de desarrollo y *product owner* y tiene como objetivo revisar las historias de usuario desarrolladas durante el *sprint*. En caso de ser aceptadas por el *product owner* serán marcadas como “hechas”. Luego de esta revisión, los participantes del evento harán una evaluación sobre los procesos llevados a cabo durante el *sprint* y como pueden ser mejorados en el siguiente.

En la Tabla 2.1 se presentan los roles de usuario identificados.

Tabla 2.1 Miembros del equipo Scrum

Rol	Nombre
Scrum <i>master</i>	Gabriel López M. Sc.
<i>Product owner</i>	Danny Guamán M. Sc. y Félix Miño
Equipo de desarrollo	Félix Miño

El proceso de desarrollo se llevó a cabo a través de 3 fases: análisis de requerimientos, diseño de la herramienta e implementación. Cada una de ellas se describe en la siguiente sección.

2.2. ANÁLISIS

Esta sección describe los procesos y artefactos Scrum utilizados para realizar el análisis de requerimientos de la herramienta.

2.2.1. REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales fueron obtenidos principalmente a través de reuniones con el *product owner*. Los requerimientos de la herramienta de anotación se resumen a continuación:

- La herramienta tendrá un mecanismo de inicio y cierre de sesión.
- La herramienta permitirá crear usuarios y asignarles un rol: anotador o administrador. Los usuarios con rol de usuario anotador son los encargados de anotar políticas de privacidad. Los usuarios con rol de usuario administrador son los encargados de administrar los elementos que conforman la herramienta y consolidar políticas de privacidad.
- La herramienta permitirá la administración de usuarios, políticas, tratamientos, atributos y valores a los usuarios con rol administrador.

- La herramienta permitirá cargar una política de privacidad representada como texto no enriquecido y separada en secciones.
- La herramienta permitirá realizar anotaciones mono o multi-valor sobre cada sección de una política de privacidad y añadir un comentario a la anotación, si es necesario.
- La herramienta permitirá la creación, visualización, edición y eliminación de las anotaciones realizadas por cada usuario (anotador o administrador) y en ningún caso se permitirá gestionar las anotaciones de otros usuarios.
- Si un anotador o administrador se encuentra en la fase de entrenamiento²⁰, la herramienta notificará al usuario de inconsistencias con anotaciones de otros anotadores.
- La herramienta calculará el nivel fiabilidad de las anotaciones realizadas por los anotadores o administradores, en la etapa de anotación, a través del coeficiente Alpha de Krippendorff [28].
- La herramienta permitirá consolidar las anotaciones realizadas por usuarios (anotador o administrador) en la etapa de anotación. Es decir en base a las anotaciones de estos usuarios fijar una anotación final que represente al conjunto de usuarios.
- La herramienta permitirá la copia de una anotación de la etapa de anotación a la etapa de consolidación.
- En la fase de consolidación, la herramienta permitirá visualizar si todas las anotaciones realizadas sobre una sección en la etapa de anotación son o no iguales.

Los requerimientos funcionales para la herramienta de visualización son los siguientes:

- La herramienta permitirá al usuario final visualizar las políticas de privacidad anotadas.

²⁰ **Fase de entrenamiento:** un usuario con rol administrador o anotador puede encontrarse en entrenamiento. Esta fase se utiliza cuando un usuario no está familiarizado con la anotación de tratamientos de datos personales y requiere ayuda adicional sobre las anotaciones que realiza sobre una política de privacidad.

- La herramienta permitirá al usuario final visualizar las anotaciones realizadas en la política de privacidad, así como los comentarios realizados.
- La herramienta permitirá al usuario final filtrar las anotaciones de acuerdo con un tratamiento de datos o atributo de interés.

2.2.2. REQUERIMIENTOS NO FUNCIONALES

- La herramienta soportará la anotación y visualización concurrente de políticas de privacidad.
- La característica de adaptabilidad es requerida en la herramienta. Es decir, que sea capaz de permitir la configuración de diferentes tipos de tratamientos de datos bajo a la estructura de tres niveles (tratamiento, atributos, y valores).
- Dado que la herramienta puede ser extendida con funcionalidades adicionales por otros desarrolladores (otros estudiantes), ésta tiene que ser implementada de tal forma que sea fácilmente mantenible (capacidad de modificación) por estos.

2.2.3. DIAGRAMA DE CASOS DE USO

La herramienta maneja tres roles de usuario: administrador, anotador y usuario final. Administrador y anotador son roles usados en la herramienta de anotación, mientras que el rol de usuario final es usado en la herramienta de visualización. El control de acceso a los módulos de la herramienta se realiza en función de los roles mencionados.

- **Administrador:** este rol de usuario permite el acceso a todos los módulos de la herramienta de anotación y es el encargado de administrar los elementos necesarios para realizar la anotación de políticas de privacidad, es decir: usuarios, políticas de privacidad, tratamientos, atributos y valores.
- **Anotador:** este rol permite el acceso solamente al módulo de anotación y es el encargado de codificar las políticas de privacidad que le fueron asignadas.
- **Usuario final:** es cualquier usuario que accede a la herramienta de visualización. Este rol de usuario tiene acceso al módulo de visualización y podrá observar las políticas de privacidad que concluyeron con el proceso de anotación.

En la Figura 2.1 y Figura 2.2 se muestran los diagramas de casos de uso para la herramienta de anotación y visualización, respectivamente. En los diagramas de caso se definen: actores que interactúan con el sistema (representado por un rectángulo), casos de uso que están representados por un óvalo, las relaciones de actores con casos de uso se

representan por una flecha y de igual manera las interacciones de un caso de uso con otro. La relación “incluye” representa la dependencia de un caso de uso con otro, mientras que “extiende” representa una relación opcional entre ellos.

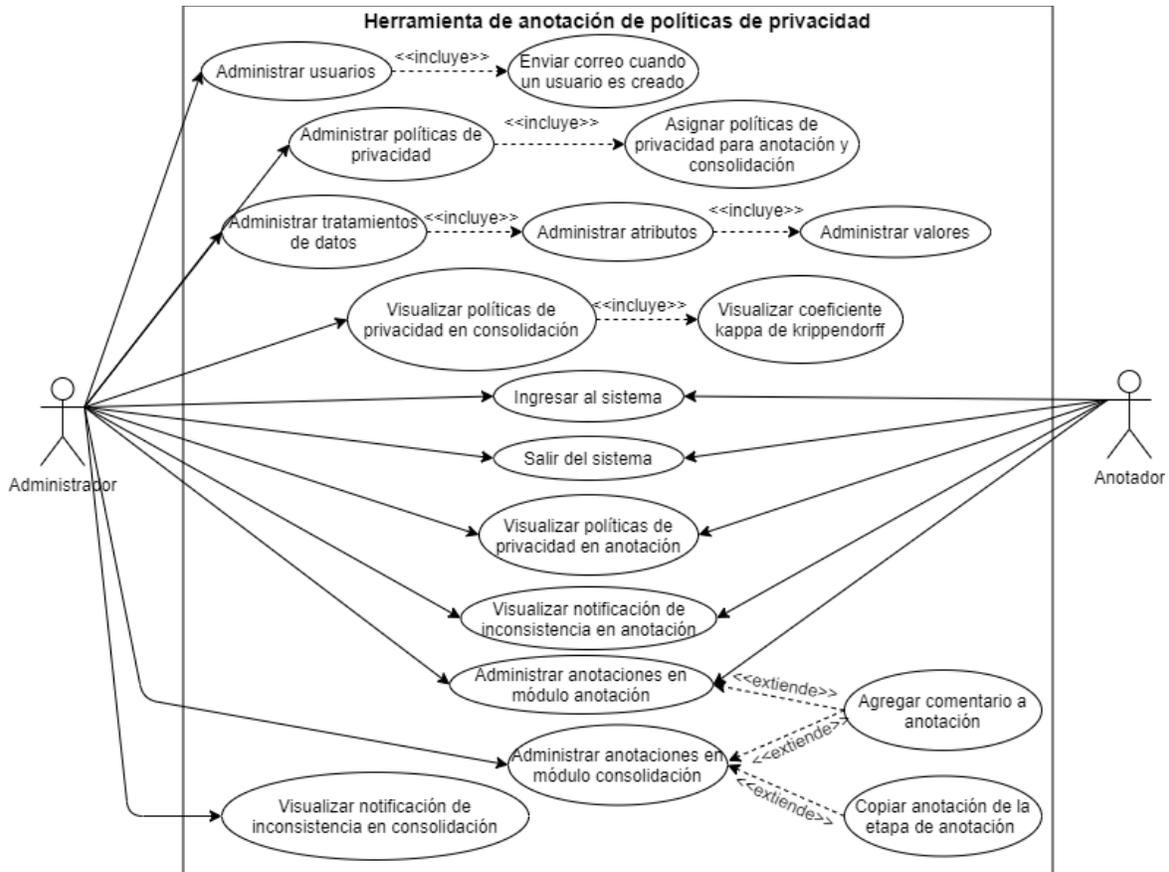


Figura 2.1 Diagrama de casos de uso de la herramienta de anotación

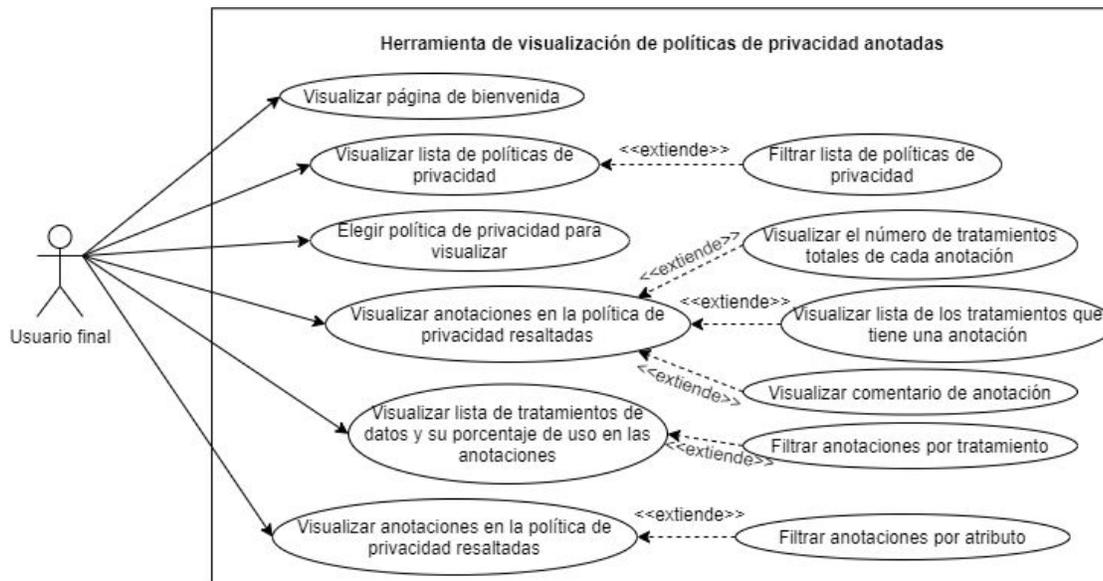


Figura 2.2 Diagrama de casos de uso herramienta de visualización

2.2.4. HISTORIAS DE USUARIO

Una vez establecidos los casos de uso, estos pueden ser traducidos a historias de usuario, que expresan de manera ordenada y detallada los requisitos con los que debe cumplir la herramienta. Todas las historias de usuario forman el *product backlog*.

En la Tabla 2.2 se observa el formato que tiene una historia de usuario:

Tabla 2.2 Formato de una historia de usuario

Nombre: Nombre descriptivo	
Código: HU-00	Asignada: Félix Miño
Prioridad: Alta, media o baja	Estimación: 1,2,3,5 u 8
Descripción: Como usuario de la herramienta (administrador, anotador o usuario final) requiero <esta funcionalidad>, de esta manera podré cumplir con <este objetivo>.	
Criterios de aceptación: <ul style="list-style-type: none">• Debe cumplir <primera condición>.• Debe cumplir <segunda condición>.• Debe cumplir <n condición>.	

Cada historia de usuario tiene:

- Nombre: un nombre que permita identificar de manera sencilla la funcionalidad a ser implementada.
- Código: identificador único y secuencial de las historias de usuario.
- Asignada: representa el miembro del equipo de desarrollo que desarrollará la historia.
- Prioridad: representa la importancia que tiene una historia de usuario de acuerdo con el criterio del *product owner*.
- Estimación: este criterio será asignado de acuerdo con lo descrito en la sección 2.1 y el equipo de desarrollo es el encargado de establecer un valor.
- Descripción: esta sección está dividida en tres partes:
 1. Como usuario de la herramienta: describe qué rol de usuario hará uso de esta funcionalidad.
 2. Requero esta funcionalidad: describe lo que necesita ser implementado en la herramienta.

3. De esta manera podré cumplir este objetivo: representa el porqué de la historia de usuario, al mismo tiempo, justifica la implementación de esta.
 - Criterios de aceptación: representan los parámetros que debe cumplir la historia de usuario para ser cambiada de la etapa de “haciendo” a la etapa de “revisión”, luego el *product owner* será el encargado de mover una historia de usuario de la etapa de “revisión” a “hecho”.

En la Tabla 2.3 se presenta como ejemplo la historia de usuario de inicio y cierre de sesión de la herramienta de anotación.

Tabla 2.3 Historia de usuario HU-01: Inicio y cierre de sesión

Nombre: Inicio y cierre de sesión	
Código: HU-01	Asignada: Félix Miño
Prioridad: Alta	Estimación: 3
<p>Descripción: Como anotador o administrador de la herramienta requiero que la herramienta de anotación cuente con un mecanismo para el inicio y cierre de sesión de esta manera se evitará accesos no autorizados.</p>	
<p>Criterios de aceptación: Como anotador o administrador requiero estar en la capacidad de:</p> <ul style="list-style-type: none"> • Iniciar sesión al ingresar mi usuario (correo electrónico) y contraseña. • Cerrar sesión. 	

La Figura 2.3 muestra un ejemplo de la historia de usuario HU-01 en la herramienta Trello.



Figura 2.3 Historia de usuario HU-01 en Trello

2.2.5. PRODUCT BACKLOG

En la Tabla 2.4 *Product Backlog* se presentan las historias de usuario organizadas en el *product backlog*. El listado completo de historias de usuario se encuentra en el ANEXO A.

Tabla 2.4 *Product Backlog*

Código	Nombre	Prioridad	Estimación ²¹
HU-01	Inicio y cierre de sesión	Alta	3
HU-02	Administración de usuarios en la herramienta	Media	5
HU-03	Administración de tratamientos de datos	Media	3
HU-04	Administración de atributos	Media	3
HU-05	Administración de valores	Media	3
HU-06	Administración de políticas de privacidad	Media	5
HU-07	Visualización de políticas de privacidad por anotar	Media	2
HU-08	Anotación de una política de privacidad	Alta	8
HU-09	Visualización, edición y eliminación de anotaciones	Media	5
HU-10	Notificación de inconsistencia en anotación	Media	5
HU-11	Visualización de políticas de privacidad para consolidación	Media	1
HU-12	Cálculo de coeficiente inter-anotador	Alta	8
HU-13	Consolidación de una política de privacidad	Media	3
HU-14	Visualización, edición y eliminación de anotaciones en consolidación	Media	3
HU-15	Visualización y copia de anotaciones de la etapa de anotación en consolidación	Media	3
HU-16	Notificación de sección inconsistente en anotación	Media	3
HU-17	Página de bienvenida en herramienta de visualización	Media	3
HU-18	Lista de políticas de privacidad para visualizar	Media	3
HU-19	Visualización de una política de privacidad anotada	Media	8
HU-20	Filtrado de anotaciones en visualización	Media	5

2.2.6. PLANIFICACIÓN DE SPRINTS

A continuación, se presenta un breve resumen de las funcionalidades a implementar en cada *sprint*.

1. Primer *Sprint* - Módulo de administración

Este módulo permitirá al usuario con rol de administrador realizar las acciones: crear, editar, visualizar y eliminar de los elementos necesarios para llevar a cabo las anotaciones. Este *sprint* contiene las historias de usuario relacionadas a la administración de los siguientes elementos: usuarios, políticas de privacidad,

²¹ **Estimación:** puede tomar uno de los siguientes valores: 1,2,3,5 u 8 que están basados en la serie de Fibonacci.

tratamientos, atributos y valores. Además, contiene la historia de usuario para el inicio y cierre de sesión, que será la primera historia por desarrollar dada su prioridad. El tiempo destinado para este *sprint* son 3 semanas.

2. Segundo *sprint* - Módulo de Anotación

Este módulo permitirá a los usuarios con rol de anotador o administrador crear, editar, visualizar y eliminar anotaciones de tratamientos de datos sobre las políticas de privacidad que les fueron previamente asignadas. Este *sprint* contiene todas las historias relacionadas con la visualización de políticas asignadas y la anotación de una política de privacidad. El tiempo destinado para este *sprint* es de 4 semanas.

3. Tercer *sprint* - Módulo de consolidación

Este módulo permitirá al usuario con rol de administrador consolidar una política de privacidad previamente anotada, es decir, en base a estas anotaciones previas fijar una anotación final que represente al conjunto de usuarios. El usuario con rol administrador podrá revisar los comentarios realizados en la etapa de anotación y realizar un comentario que será el que el usuario final visualiza.

Así mismo, este módulo calculará el nivel de concordancia inter-anotador haciendo uso del coeficiente Alpha de Krippendorff, que permitirá conocer el grado de similitud que existe en las anotaciones realizadas por los usuarios (anotador o administrador) sobre la política de privacidad, en la etapa de anotación. El tiempo destinado para este *sprint* es de 4 semanas.

4. Módulo de visualización

Este módulo será público ya que su objetivo es que los usuarios finales puedan ingresar a la herramienta y visualizar las políticas de privacidad anotadas. Las anotaciones de las políticas de privacidad y los comentarios que se presentan en este módulo corresponden a la etapa de consolidación. Se podrá filtrar las anotaciones por tratamiento de datos y por alguno de los atributos que componen este tratamiento. El tiempo destinado para este *sprint* será de 4 semanas,

Se organizaron las historias en diferentes *sprints* de manera que cada uno cubra un módulo en su totalidad.

En la Tabla 2.5 se muestran las historias de usuario que contiene cada *sprint*.

Tabla 2.5 Planificación de *sprints*

#	<i>Sprint</i>	Historia de usuario			
		Código	Nombre	Prioridad	Estimación
1	Módulo de administración	HU-01	Inicio y cierre de sesión	Alta	3
		HU-02	Administración de usuarios en la herramienta	Media	5
		HU-03	Administración de tratamientos de datos	Media	3
		HU-04	Administración de atributos	Media	3
		HU-05	Administración de valores	Media	3
		HU-06	Administración de políticas de privacidad	Media	5
Total de puntos del <i>sprint</i>					22
2	Módulo de anotación	HU-07	Visualización de políticas de privacidad por anotar	Media	2
		HU-08	Anotación de una política de privacidad	Alta	8
		HU-09	Visualización, edición y eliminación de anotaciones	Media	5
		HU-10	Notificación de inconsistencia en anotación	Media	5
Total de puntos del <i>sprint</i>					20
3	Módulo de consolidación	HU-11	Visualización de políticas de privacidad para consolidación	Media	1
		HU-12	Cálculo de coeficiente inter-anotador	Alta	8
		HU-13	Consolidación de una política de privacidad	Media	3
		HU-14	Visualización, edición y eliminación de anotaciones en consolidación	Media	3
		HU-15	Visualización y copia de anotaciones de la etapa de anotación en consolidación	Media	3
		HU-16	Notificación de sección inconsistente en anotación	Media	3
Total de puntos del <i>sprint</i>					21
4	Módulo de visualización	HU-17	Página de bienvenida en herramienta de visualización	Media	3
		HU-18	Lista de políticas de privacidad para visualizar	Media	3
		HU-19	Visualización de una política de privacidad anotada	Media	8
		HU-20	Filtrado de anotaciones en visualización	Media	5
Total de puntos del <i>sprint</i>					19

La Figura 2.4 muestra como ejemplo el desarrollo del *sprint* 1 en Trello²².

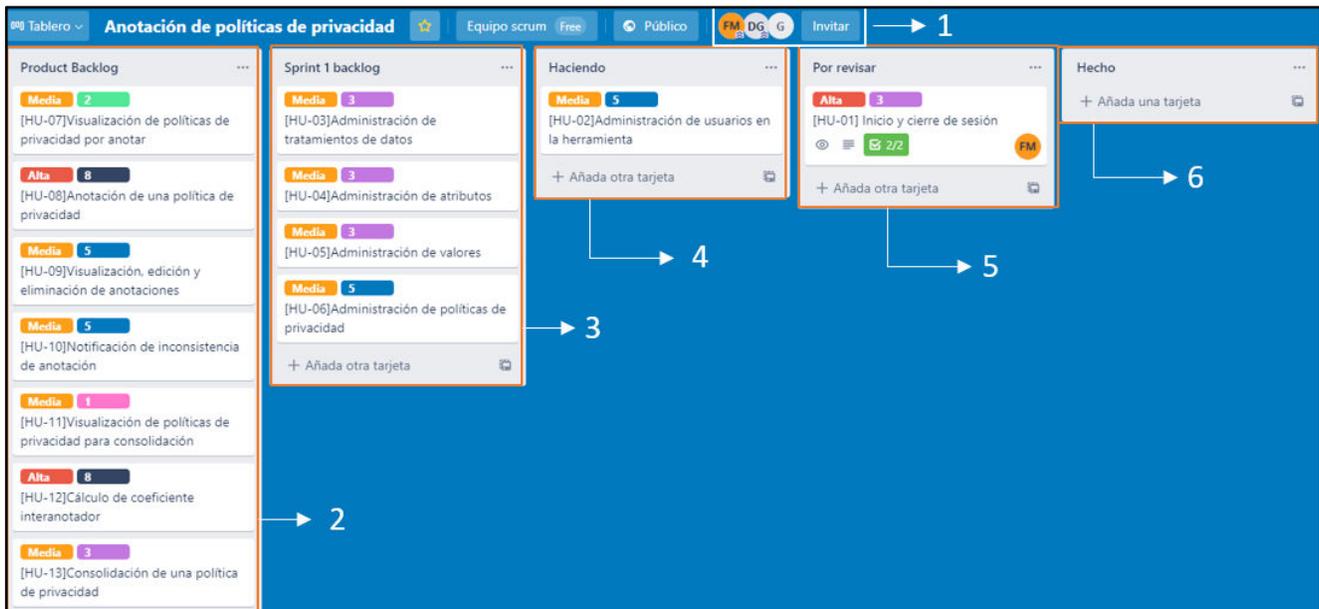


Figura 2.4 *Sprint* 1 en Trello

En la Figura 2.4 se encuentran los siguientes elementos:

1. Equipo Scrum: *Trello* es una herramienta colaborativa y permite trabajar en conjunto con el equipo Scrum.
2. *Product backlog*: es donde se encuentran las historias de usuario priorizadas.
3. *Sprint backlog*: historias de usuario correspondientes al *sprint* en desarrollo.
4. Haciendo: se muestra la historia de usuario en desarrollo. En esta columna sólo se tiene una historia a la vez.
5. Por revisar: historias de usuario realizadas y listas para ser analizadas en la reunión revisión del *sprint*.
6. Hecho: luego de la reunión de revisión/retrospectiva del *sprint* el *product owner* decidirá si una historia de usuario cumple con los requerimientos iniciales.

2.3. DISEÑO

En esta sección se presenta el diseño de la herramienta. El esquema inicial que guió el diseño de la herramienta se presenta en la Figura 2.5. La herramienta esta compuesta por:

²² Dirección web de tablero Trello usado para el desarrollo de la herramienta:
<https://trello.com/b/N8Ntg6om/ anotaci%C3%B3n-de-pol%C3%ADticas-de-privacidad>

- Dos aplicaciones web: herramienta de anotación y herramienta de visualización. Conforman el *frontend*²³ de la herramienta.
- Una aplicación de servidor que se comunica con la base de datos y ejecuta las acciones solicitadas por las aplicaciones web, *backend*²⁴ de la herramienta.

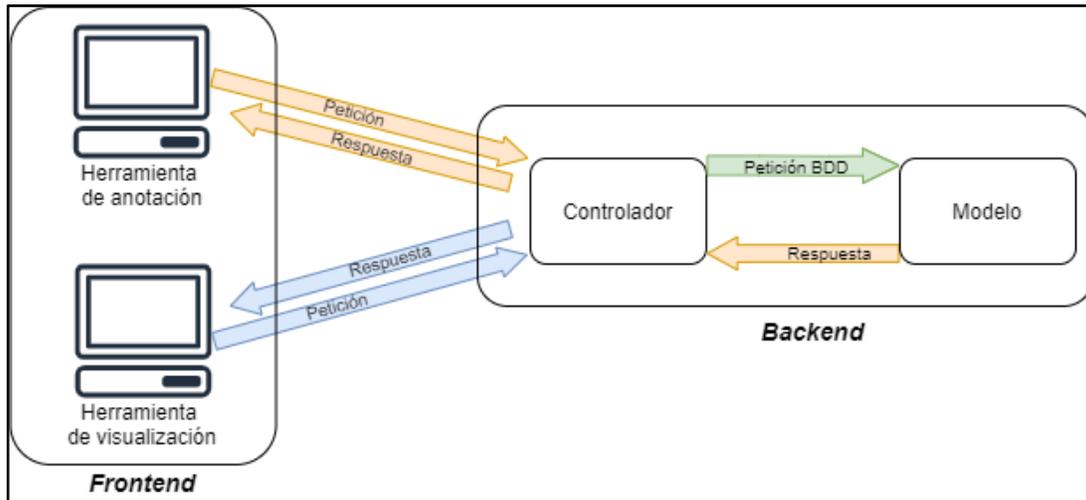


Figura 2.5 Diagrama general de la herramienta

2.3.1. ESTRUCTURA DE UNA ANOTACIÓN

A continuación, se realiza una explicación de la estructura de anotación con el objetivo de aclarar este término para siguientes secciones. En la Figura 2.6 se presenta la estructura de una anotación.

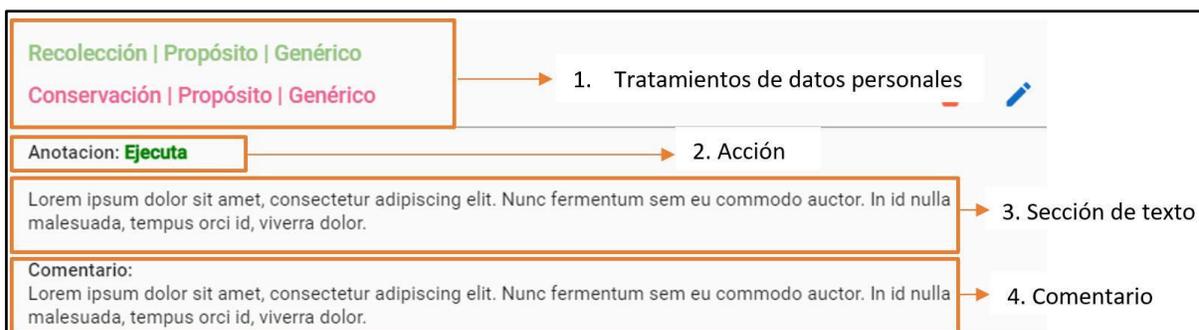


Figura 2.6 Estructura de una anotación

Una anotación está conformada por:

1. Uno o más tratamientos de datos personales.

²³ **Frontend:** es la parte visual de una aplicación que interactúa con el usuario.

²⁴ **Backend:** es la parte de un sistema informático que se encarga de la manipulación y almacenamiento de datos.

2. Un valor booleano que indica si los tratamientos se ejecutan o no sobre los datos personales (1 = **ejecuta** y 0 = **no ejecuta**).

Por ejemplo, de la política de privacidad de Facebook se extrajeron los siguientes segmentos²⁵: 1) “No obstante, a menos que nos des permiso para hacerlo, **no compartimos información** que te identifique personalmente, como tu nombre o dirección de correo electrónico, datos que se puedan emplear para ponerse en contacto contigo o que puedan revelar tu identidad.”, este ejemplo **no ejecuta** el tratamiento y 2) “**Recopilamos** el contenido, las comunicaciones y otros datos que proporcionas cuando usas nuestros Productos, por ejemplo, cuando te registras para crear una cuenta, creas o compartes contenido y envías mensajes a otras personas o te comunicas con ellas” el cual corresponde a un ejemplo donde el tratamiento de recolección de datos se **ejecuta**.

3. Una sección de texto.
4. Un comentario (opcional).

2.3.2. MODELO – DISEÑO DE BASE DE DATOS

En esta sección se presentan los diagramas entidad-relación y relacional que permiten observar de manera gráfica los distintos elementos que contiene la base de datos de la herramienta y la manera en que estos interactúan.

El modelo entidad-relación también llamado modelo conceptual, permite representar de manera gráfica los objetos, llamados entidades, y las relaciones entre estas entidades, que en conjunto forman una base de datos. En la Figura 2.7 se presenta el diagrama entidad-relación de la herramienta.

Como ejemplo para entender de mejor manera el diagrama entidad-relación se toma la entidad *Anotación* ya que aquí convergen la mayoría de las entidades. Los atributos de esta entidad son:

- **Id**: identificador único de una anotación.
- **Texto**: representa el texto que fue seleccionado.
- **Texto HTML**: es el texto seleccionado con ciertas etiquetas html para que pueda ser presentado de manera correcta en la herramienta de visualización.

²⁵ <https://es-es.facebook.com/privacy/explanation>

- **Comentario:** este campo es opcional y representa el comentario que el usuario (anotador o administrador) realiza sobre una anotación.
- **Consolidar:** este campo es de tipo booleano y representa si una anotación es de tipo consolidación o anotación. Una anotación de tipo consolidación es la que se presenta en la herramienta de visualización, mientras que una de tipo anotación es presentada como guía para el usuario asignado a consolidar la política de privacidad.
- **Ejecuta:** este campo es de tipo booleano y representa si los tratamientos de datos se ejecutan o no en la anotación.
- **Fecha anotado:** es una marca de tiempo del momento de realización de la anotación.

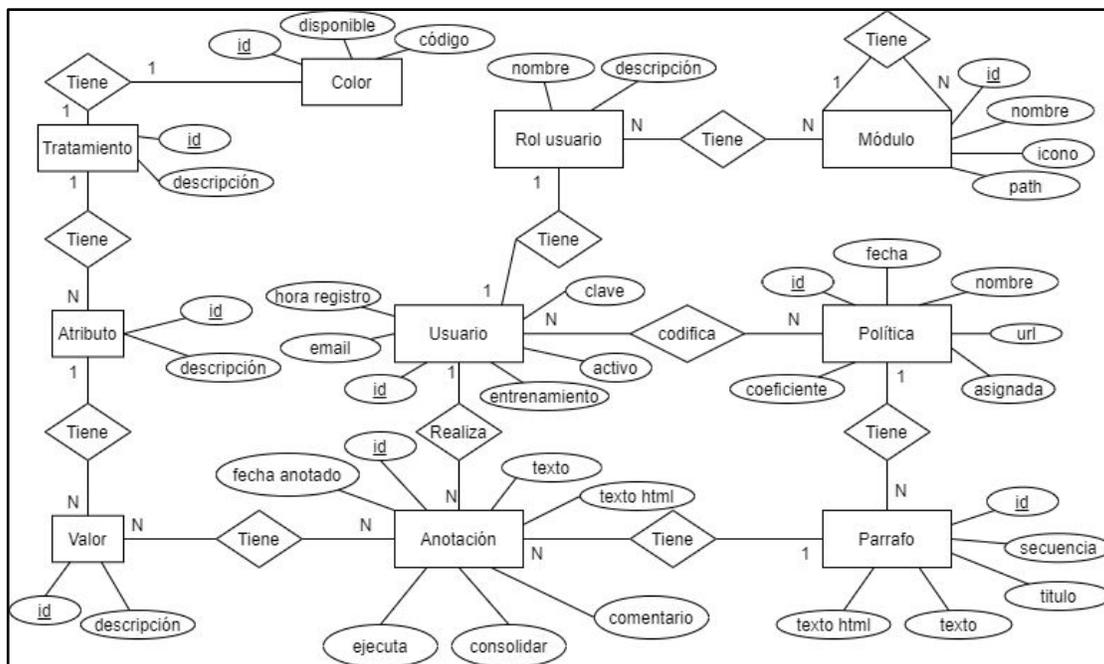


Figura 2.7 Diagrama entidad-relación

Las relaciones que tiene esta entidad son las siguientes:

- Un Usuario (anotador o administrador) puede realizar cero, una o varias Anotaciones en las etapas de anotación o consolidación. Cero anotaciones significan que un tratamiento de datos no está contenido en la sección anotada.
- Un Párrafo puede tener cero, una o varias Anotaciones.
- Una Anotación puede tener uno o varios Valores.

De la misma manera, en la Figura 2.8 se presenta el diagrama relacional. En un diagrama relacional los datos y las relaciones entre los datos se presentan por medio de un conjunto de tablas, cada tabla tiene varias columnas y se denomina atributos, mientras que una fila de una tabla es un registro. El diseño de una base de datos se denomina esquema.

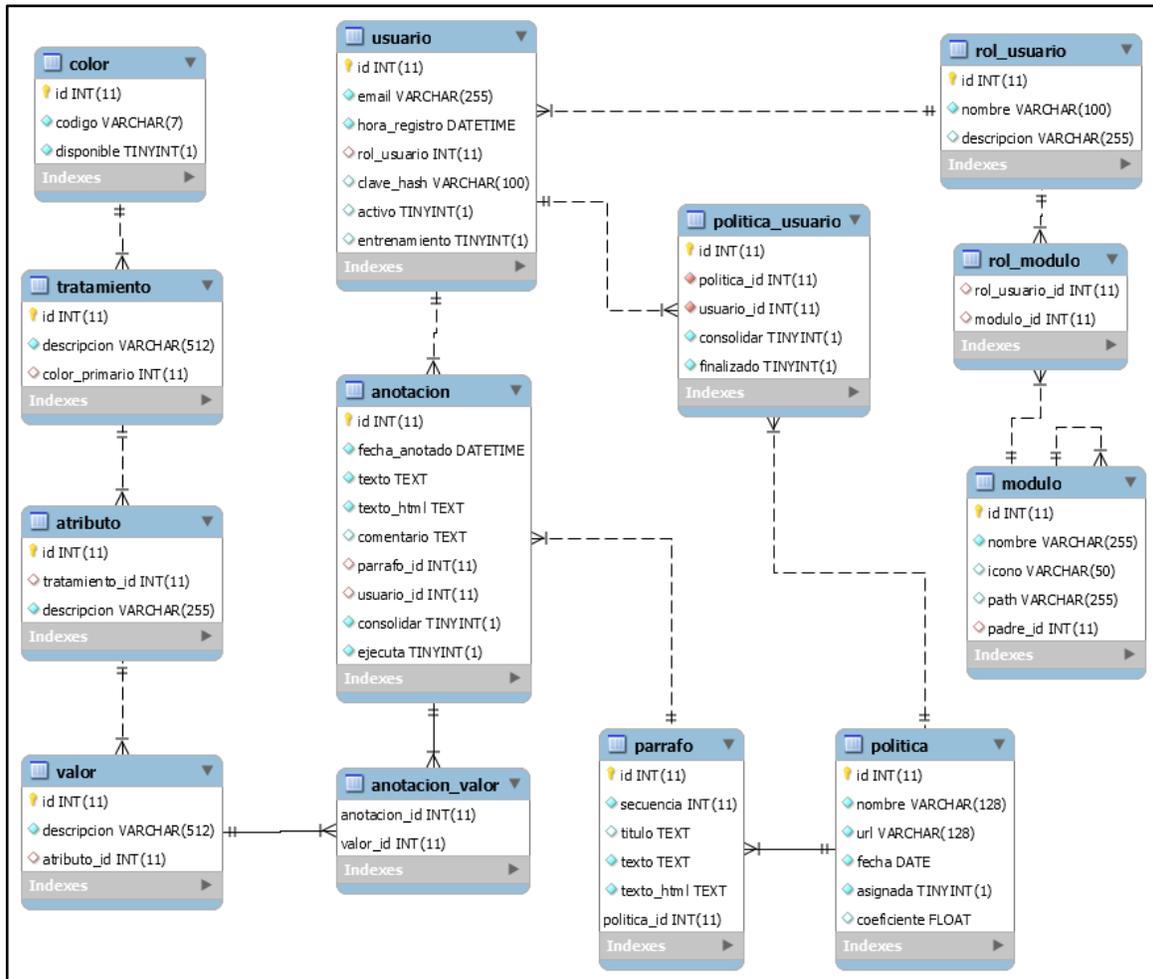


Figura 2.8 Diagrama relacional

A continuación, se explica de manera breve las tablas que componen la base de datos:

- **usuario**: esta tabla almacena todos los usuarios que se encuentran en la herramienta. Se encuentran usuarios del tipo anotador y administrador.
- **rol_usuario**: esta tabla tiene los roles de usuario permitidos en la herramienta de anotación, se permiten los roles: anotador y administrador.
- **modulo**: esta tabla almacena los módulos y submódulos de la herramienta. Es una tabla auto relacionada, es decir que un módulo puede tener como padre un módulo

de la misma tabla, cualquier módulo que tenga un módulo padre se denominará submódulo. Esta tabla funciona como una vista de árbol ²⁶.

- `rol_modulo`: esta tabla nace de la relación de `rol_usuario` y `modulo`. Permite relacionar un rol de usuario con un `modulo` en particular. En esta tabla no se relaciona un `rol_usuario` con un submódulo.
- `politica`: esta tabla almacena las políticas de privacidad que serán anotadas en la herramienta.
- `politica_usuario`: esta tabla permite relacionar una política de privacidad con un usuario, es utilizada para la asignación de políticas en la etapa de anotación y consolidación.
- `parrafo`: esta tabla almacena los párrafos o secciones de una política de privacidad que fue subida a la herramienta en formato `txt`.
- `color`: esta tabla almacena los colores que se podrán asignar a un tratamiento de datos para realizar el resaltado.
- `tratamiento`: esta tabla almacena los tratamientos de datos que se identificaron en el PLOPDP para la anotación de políticas de privacidad.
- `atributo`: esta tabla almacena los atributos de un tratamiento.
- `valor`: esta tabla almacena los valores de un atributo.
- `anotacion`: esta tabla almacena toda la información relacionada a la anotación como es: `fecha_annotado`, `texto`, `texto_html`, un comentario, el id de párrafo, el id del usuario que realiza la anotación, si la anotación pertenece a la etapa de anotación o de consolidación y
- `anotacion_valor`: esta tabla almacena el o los valores, que a su vez se relacionan con un atributo y éste con un tratamiento, que se expresan en una anotación.

Con el objetivo de cumplir con el requerimiento no funcional que busca que la mantenibilidad de la herramienta se decidió usar el ORM flask-SQLAlchemy el cual permite extraer toda la lógica de la base de datos al lenguaje de programación Python.

²⁶ **Vista de árbol**: representa una vista jerárquica de información.

En el Código 2.1 se presenta el ejemplo de una consulta ejecutada con SQL y con Flask-SQLAlchemy, con el fin de comparar estos métodos de acceso a la base de datos.

```


Sentencia SQL



```
SELECT COUNT(*) FROM anotacion
LEFT OUTER JOIN parrafo ON anotacion.parrafo_id = parrafo.id
LEFT OUTER JOIN politica ON parrafo.politica_id = politica.id
WHERE politica.id = ?
AND anotacion.usuario_id = ?
AND anotacion.consolidar =?;
```



```
(db.session.query(Anotacion)
.outerjoin(Parrafo, Anotacion.parrafo_id == Parrafo.id)
.outerjoin(Politica, Parrafo.politica_id == Politica.id)
.filter(Politica.id == politica_id,
 Anotacion.usuario_id == usuario_id,
 Anotacion.consolidar == consolidar).count())
```



Sentencia flask-SQLAlchemy


```

Código 2.1 Consulta ejecutada con SQL y Flask-SQLAlchemy

La consulta ejecutada en el Código 2.1 retorna el conteo de todos los valores de las anotaciones de consolidación de una política de privacidad. Como se puede observar una consulta ejecutada con Flask-SQLAlchemy (recuadro en verde) es similar a la ejecución de una función o método de una clase. Durante la implementación de este trabajo se usará este estilo de consultas con Flask-SQLAlchemy.

2.3.3. VISTA – DISEÑO DE INTERFACES

Las interfaces visuales, *mockups*²⁷, fueron diseñadas con la herramienta Adobe XD²⁸. Los *mockups* se usaron como guías para definir la estructura de las interfaces e interacción del usuario con la herramienta y no para ser seguidos de manera exacta. Todas las interfaces fueron diseñadas para una resolución FHD (Full High Definition, 1920x1080px). Es importante mencionar que las interfaces están optimizadas para el navegador Google Chrome, queda fuera del alcance de este trabajo la optimización para otros navegadores.

En la Figura 2.9 se presenta la interfaz de consolidación, en la Figura 2.10 se presenta la interfaz de anotación y la Figura 2.11 corresponde a la herramienta de visualización. Las interfaces de todos los módulos se presentan en el ANEXO B.

En la Figura 2.9 se presenta la interfaz de consolidación de una política de privacidad. El menú lateral presentado corresponde a un usuario con rol administrador. La sección

²⁷ **Mockup**: es la fotografía de una interfaz gráfica, permiten al *product owner* visualizar las interfaces gráficas antes de ser implementadas en código.

²⁸ <https://www.adobe.com/la/products/xd.html>

izquierda superior corresponde al texto de la sección a anotar, la sección izquierda inferior corresponde a los tratamientos de datos que se encuentran disponibles para la anotación de la política de privacidad y la sección derecha contiene las anotaciones realizadas en la etapa de anotación e indica si éstas son iguales

En la Figura 2.10 se presenta la interfaz de anotación. El menú lateral presenta solamente la opción “Anotación” ya que es la interfaz que tendrá un usuario con rol anotador. La sección izquierda de la interfaz corresponde al texto de la sección que se está anotando y la sección derecha corresponde a los tratamientos de datos que se encuentran disponibles para la anotación de la política de privacidad.

En la Figura 2.11 se muestra la interfaz de visualización de una política de privacidad anotada. Es importante mencionar dos características: 1) el color con el que se resalta una anotación es el correspondiente al color del primer tratamiento anotado y 2) el color de relleno de los tratamientos a la izquierda es el porcentaje que cada tratamiento representa del total de anotaciones en la política. La sección izquierda de la interfaz corresponde a todos los tratamientos de datos que se encuentran disponibles para la anotación de una política de privacidad y la sección derecha corresponde a la política de privacidad resaltada por los tratamientos de datos que en ella han sido anotados.

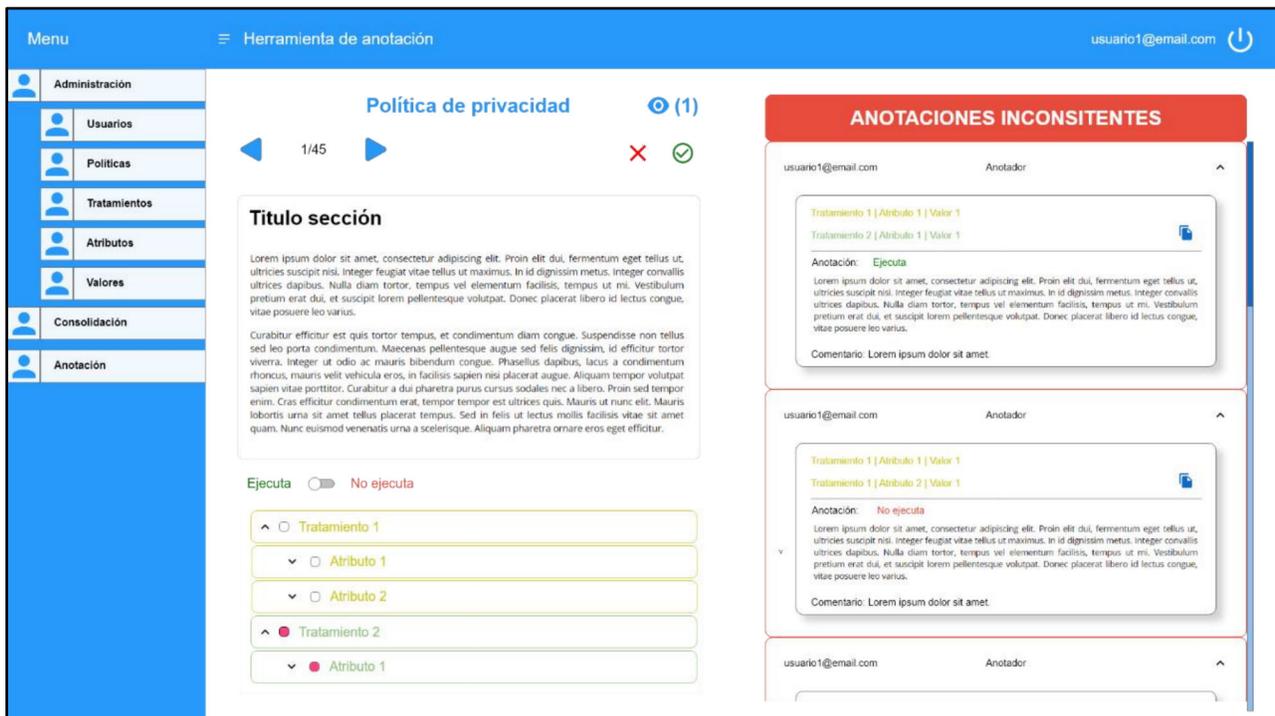


Figura 2.9 Interfaz de consolidación (párrafo/sección inconsistente)

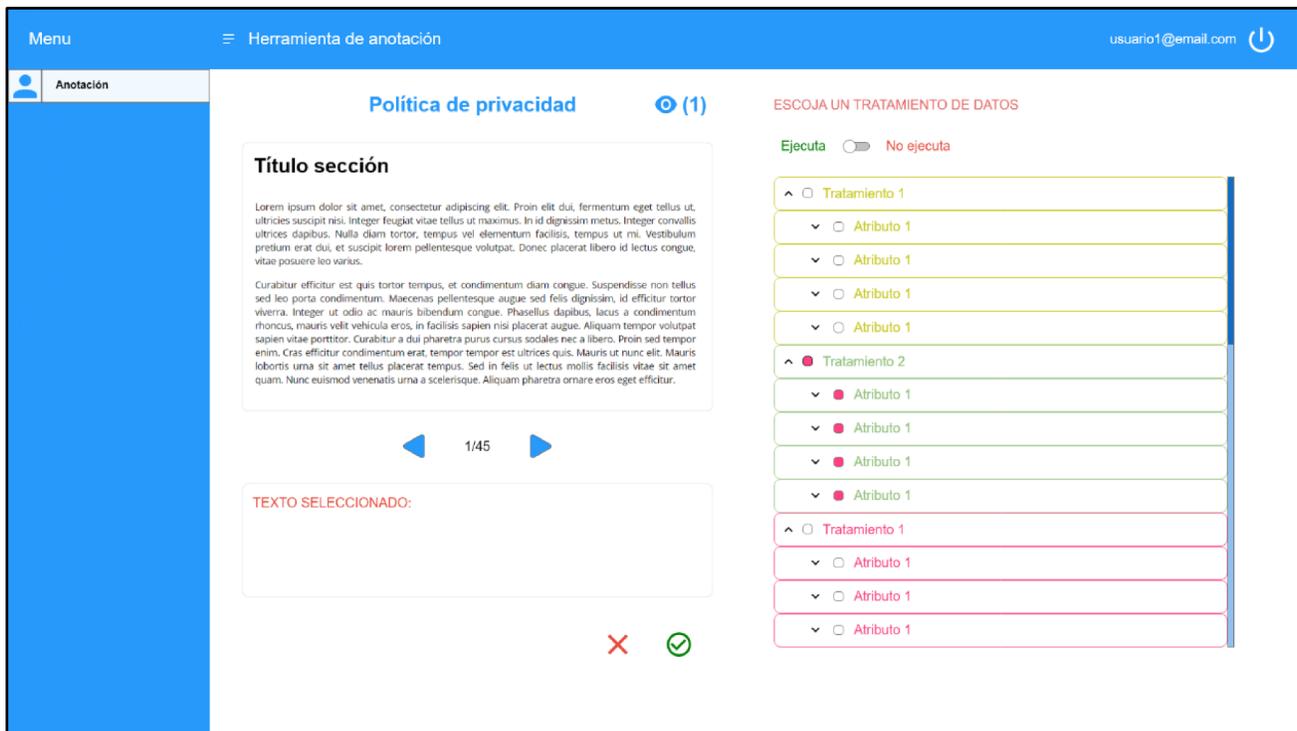


Figura 2.10 Interfaz de anotación



Figura 2.11 Interfaz de visualización de política de privacidad anotada

2.3.4. CONTROLADOR – DISEÑO DE CLASES

En esta sección se presenta el diagrama de clases y los diagramas de flujo de las funcionalidades más relevantes de la herramienta. Un diagrama de clases representa los aspectos estáticos de las clases que componen la herramienta. Cada objeto del mundo

real es abstraído en componentes llamados clases. Cada clase posee atributos y métodos que determinan su comportamiento. En la Figura 2.12 se presenta el diagrama de clases de la herramienta.

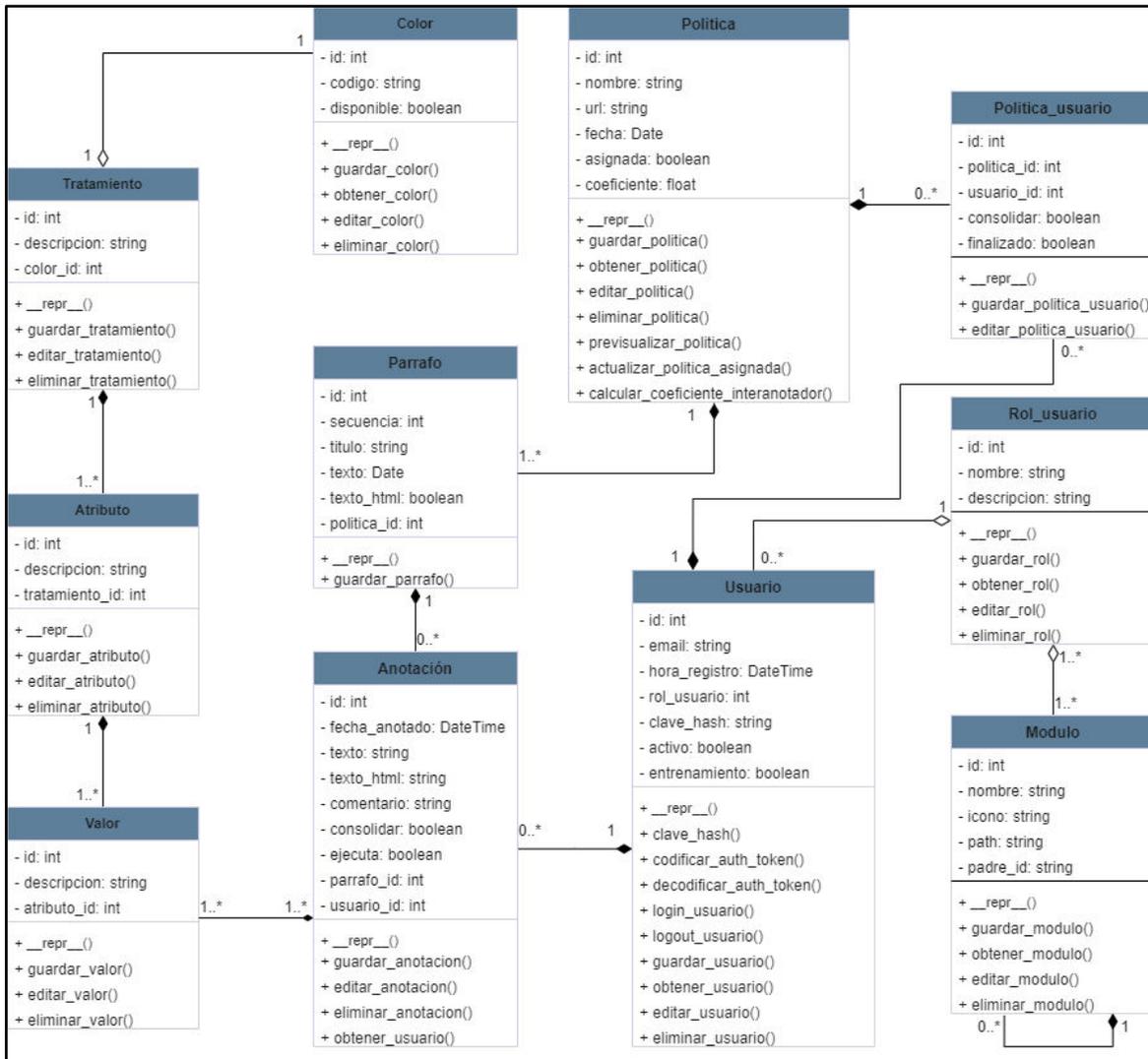


Figura 2.12 Diagrama de clases de la herramienta

Las clases fueron nombradas igual que las entidades presentadas en el diagrama relacional en la Figura 2.8, los atributos de una clase corresponden a columnas en la base de datos y los métodos ejecutan acciones sobre sus datos.

Por ejemplo, para la clase `Color` sus atributos son:

- `id`: corresponde a la llave primaria en la base de datos y es un valor numérico.
- `codigo`: corresponde al código de color en notación hexadecimal y su tipo de datos

- `disponible`: es un valor booleano que indica si un color está disponible para ser asignado a un tratamiento de datos.

Los métodos implementados para la clase `Color` son también usados en otras clases, a continuación se explica su funcionalidad:

- `__repr__`: es un método usado para la representación textual de un objeto. Principalmente es usado con fines de depuración.
- `guardar()`: este método permite guardar un objeto de una clase en una tabla en la base de datos.
- `obtener()`: recupera un registro o una lista de registros desde la base de datos.
- `editar()`: modifica los valores de un registro en la base de datos.
- `eliminar()`: remueve un registro de la base de datos.

En cuanto a las relaciones que tiene la clase `Color`:

- un `Tratamiento` tiene un `Color` asignado.

Por otro lado, un diagrama de flujo es un diagrama que describe un proceso de manera gráfica. Se usan ampliamente en numerosos campos para documentar procesos que suelen ser complejos en diagramas claros y fáciles de comprender.

A continuación, se presentan los diagramas de flujo de las siguientes funcionalidades de la herramienta:

- Notificación de inconsistencia en anotación (Figura 2.13).
- Cálculo de coeficiente inter-anotador (Figura 2.14).
- Notificación de sección inconsistente en consolidación (Figura 2.15).
- Presentación de política de privacidad anotada (Figura 2.16).
- Filtrado de anotaciones en módulo visualización (Figura 2.17).

La Figura 2.13 presenta el diagrama de flujo para la notificación de inconsistencia en anotación. Esta característica de la herramienta permite a usuarios con rol de anotador o administrador, que se encuentren en entrenamiento, ser notificados de posibles errores que se estén cometiendo en la etapa de anotación. La notificación está basada en anotaciones que otros usuarios realizaron previamente sobre la política de privacidad.

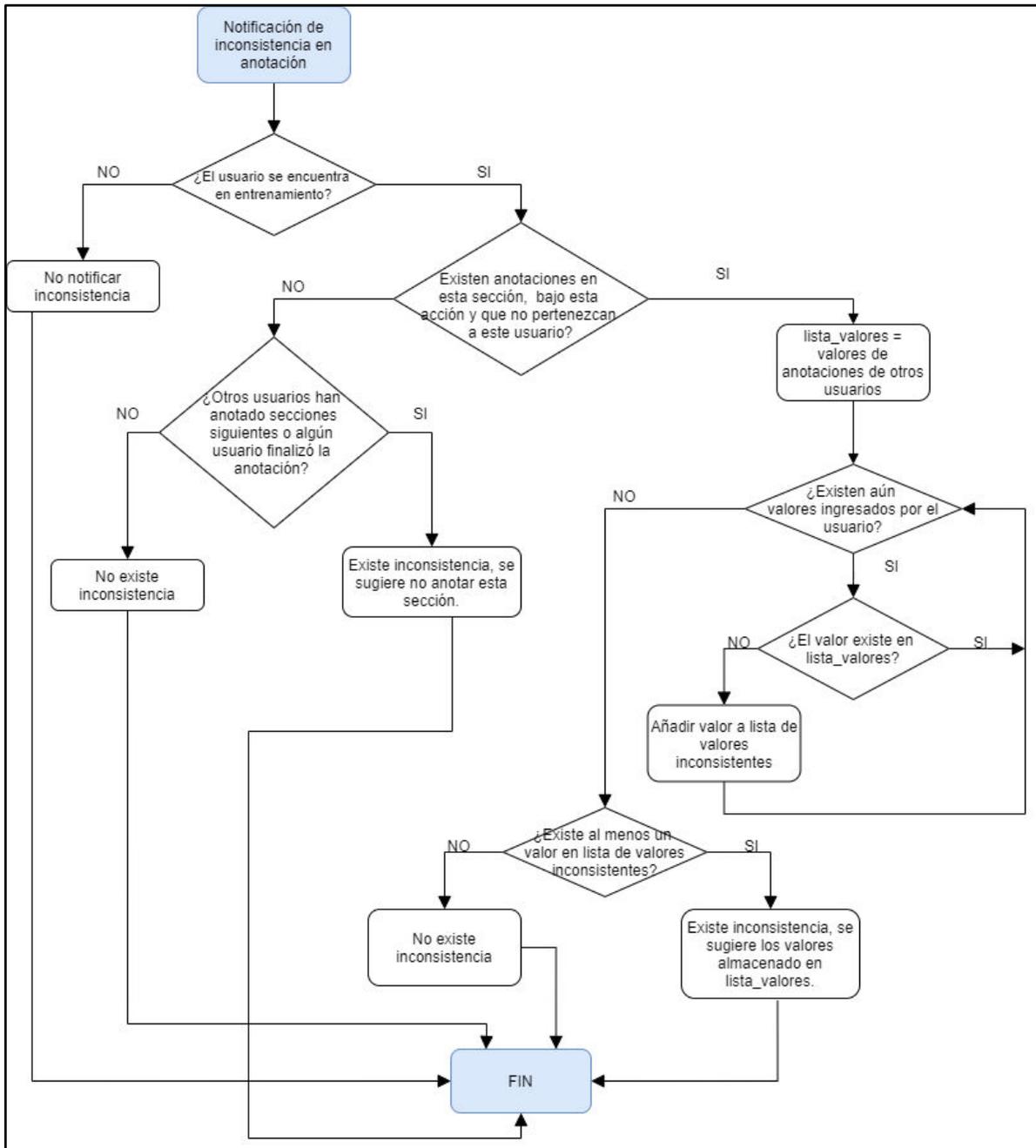


Figura 2.13 Diagrama de flujo de notificación de inconsistencia en anotación

La Figura 2.14 presenta el diagrama de flujo para el cálculo de coeficiente inter-anotador. Esta característica de la herramienta define, en la etapa de consolidación, el nivel de fiabilidad que se puede tener respecto a las anotaciones realizadas en la etapa de anotación.

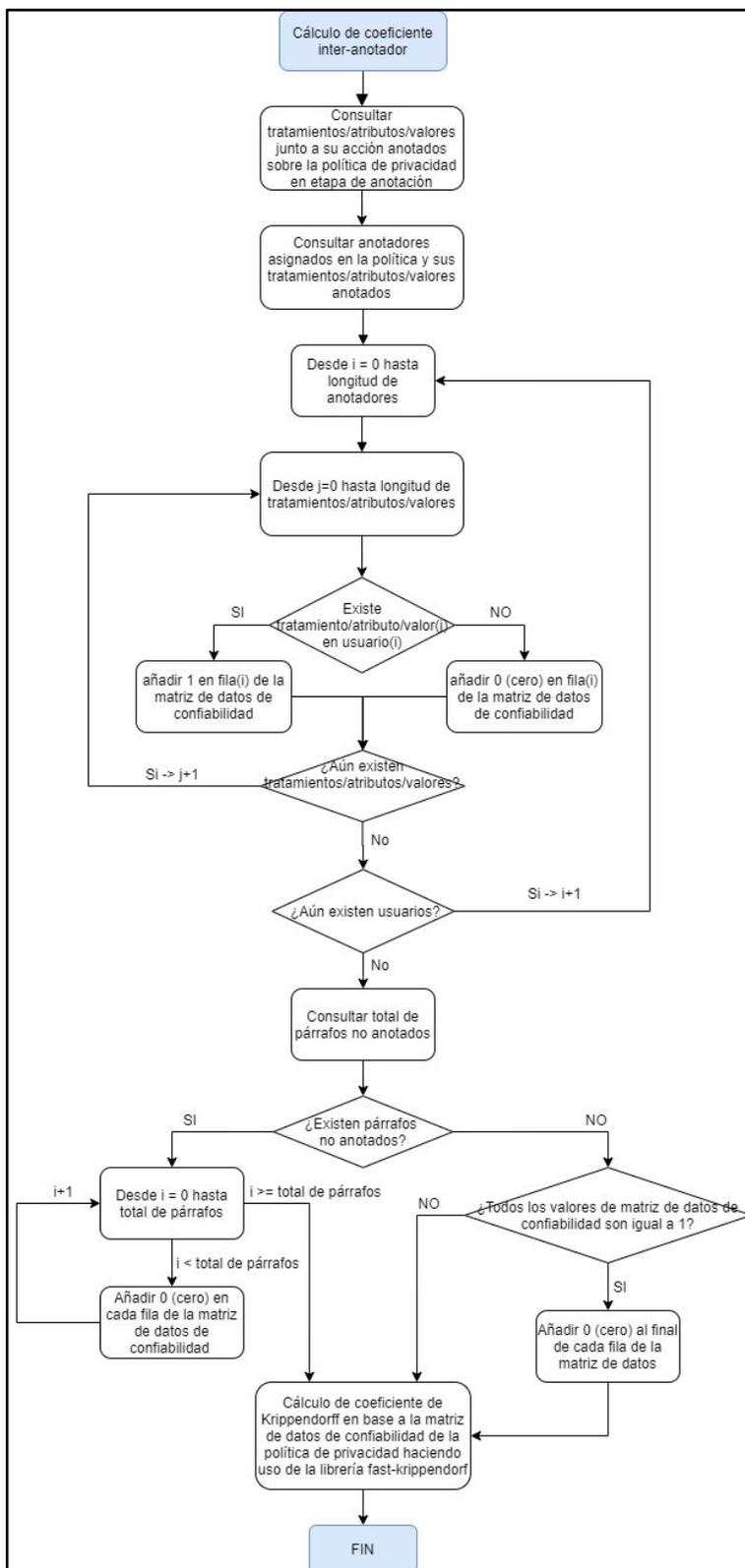


Figura 2.14 Diagrama de flujo de cálculo de coeficiente inter-annotador

La Figura 2.15 presenta el diagrama de flujo para la notificación de inconsistencia en consolidación. Esta característica permite al usuario que consolida una política de

privacidad tener una notificación gráfica de la consistencia de las anotaciones de cada sección. Existen 3 casos:

1. Una sección es consistente porque todas las anotaciones de los anotadores son iguales (sin tomar en cuenta el texto anotado).
2. Una sección es consistente porque ningún anotador realizó anotaciones sobre ésta.
3. Una sección es inconsistente porque sus anotaciones no son iguales.

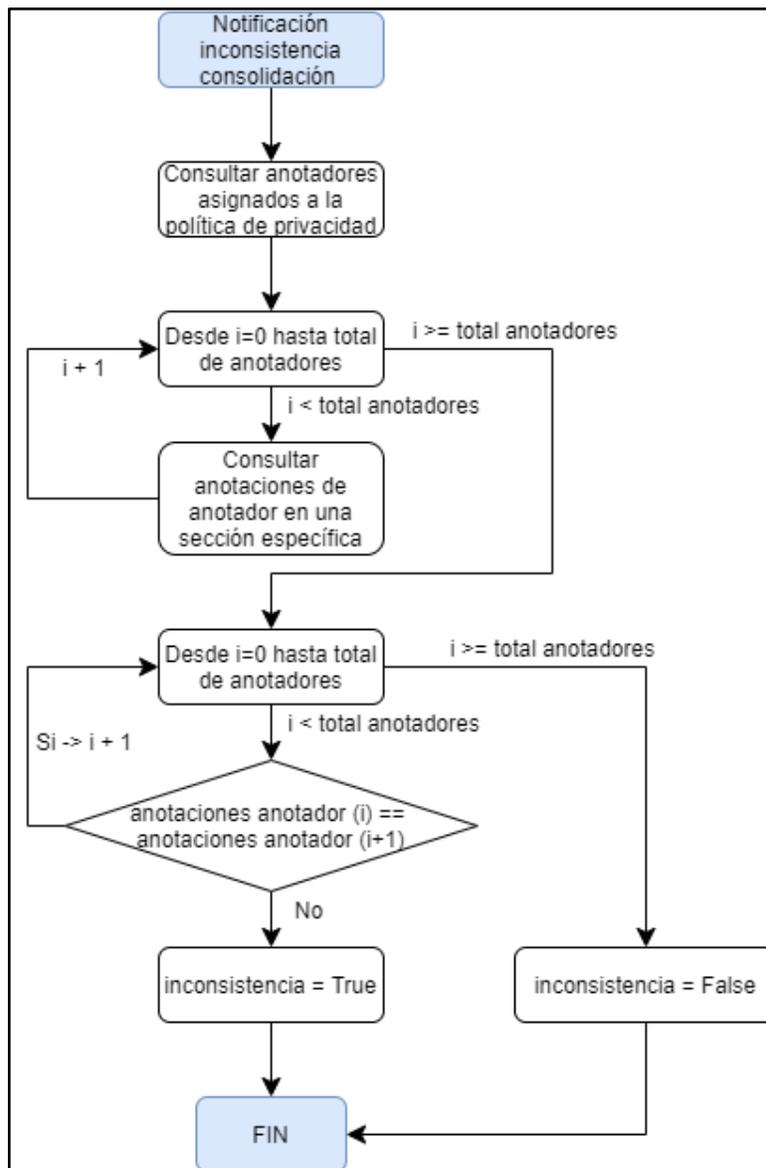


Figura 2.15 Diagrama de flujo notificación de sección inconsistente en consolidación

La Figura 2.16 presenta el diagrama de flujo de la presentación de una política de privacidad en la herramienta de visualización. En este proceso es donde se incluye los estilos (CSS) necesarios para estructurar una política de privacidad.

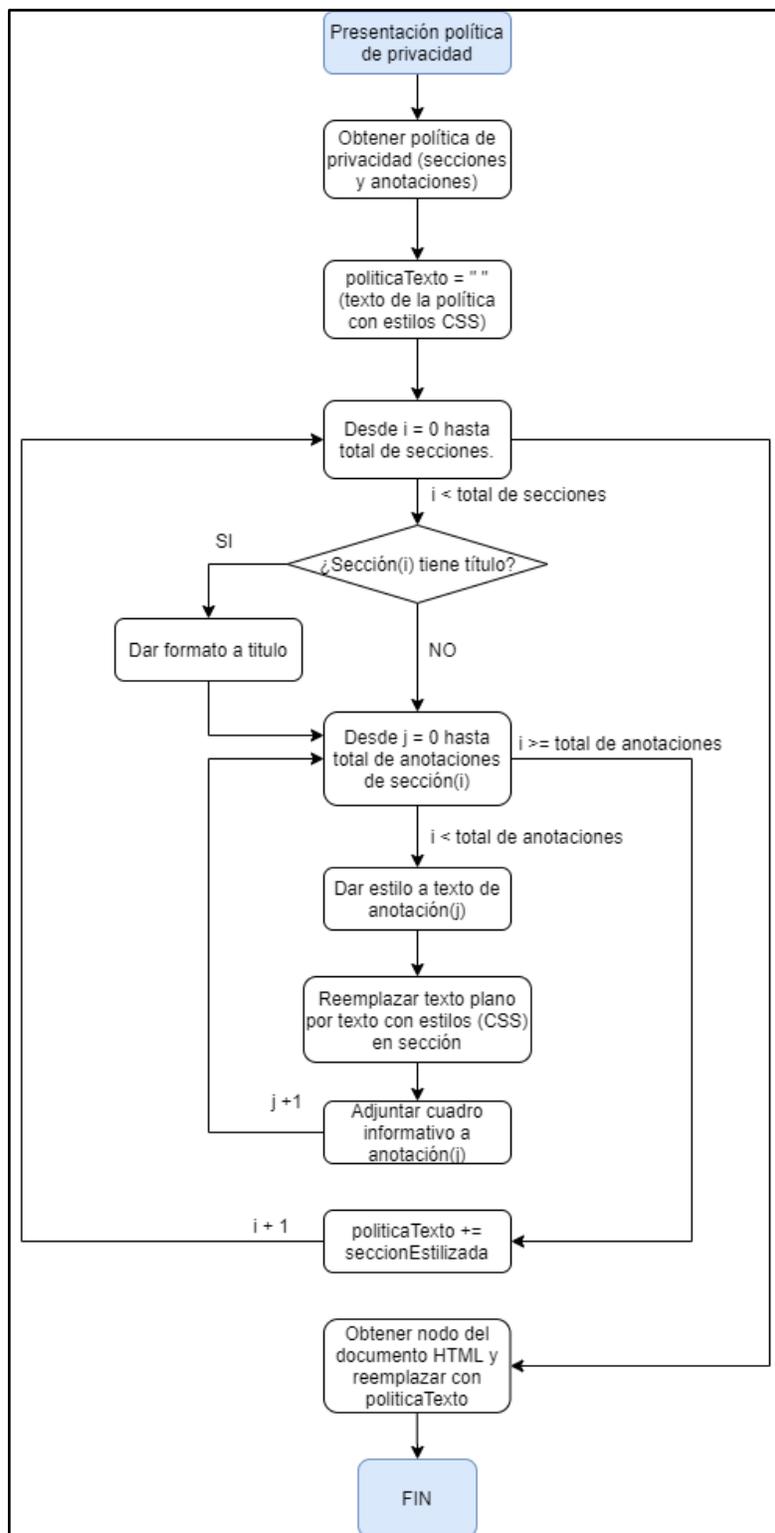


Figura 2.16 Diagrama de flujo de presentación política de privacidad anotada

Finalmente, la Figura 2.17 presenta el diagrama de flujo de las interacciones del usuario final con los filtros de tratamiento de datos y atributo en la herramienta de visualización. Se presenta la lógica que se ejecuta cuando el usuario selecciona alguno de estos filtros.

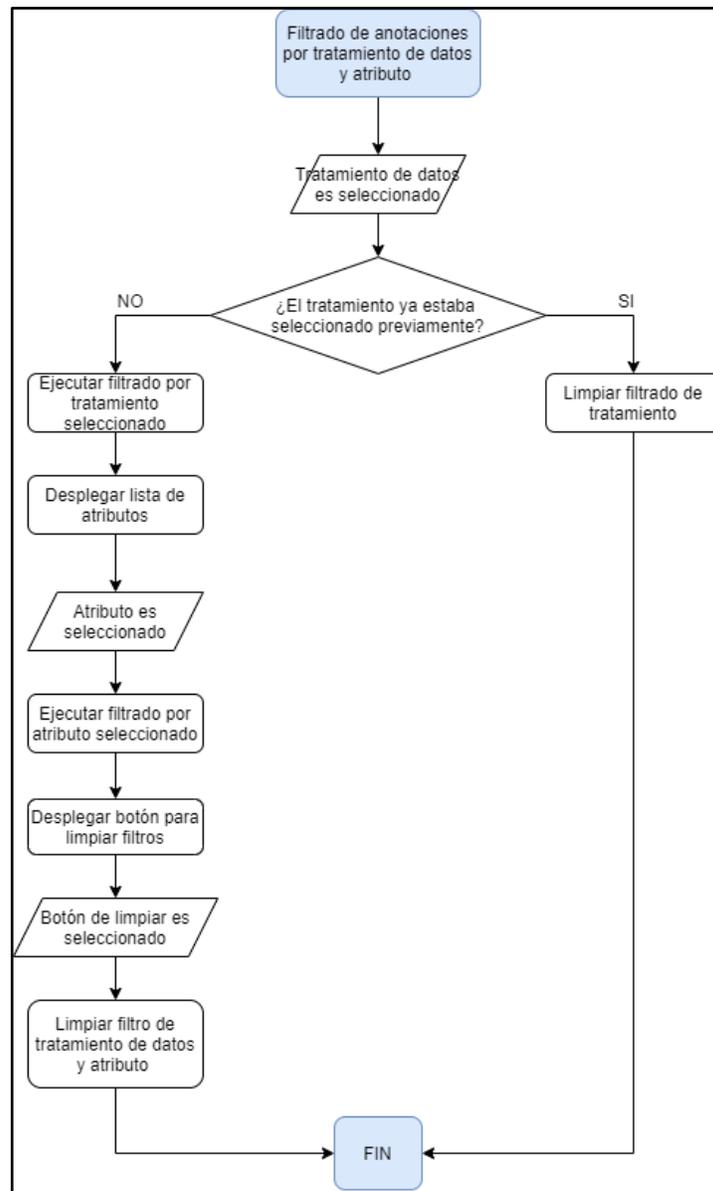


Figura 2.17 Diagrama de flujo filtrado de anotaciones en módulo visualización

2.4. IMPLEMENTACIÓN

En esta sección se usaron los lenguajes de programación explicados en la sección 1.3.7. Los componentes de la herramienta fueron implementados con las versiones de los lenguajes de programación y herramientas que se presentan en la Tabla 2.6.

En cuanto a la comunicación de la vista con el controlador se usó APIs RESTful. REST (Representational State Transfer) es una arquitectura para el diseño de servicios web que tiene por objetivo comunicar datos o ejecutar operaciones sobre estos [29]. Una petición a un servicio web está compuesta por:

- URI (*Uniform Resource Identifier*): identifica de manera única un servicio web.

- Método HTTP: Es la acción que se llevará a cabo sobre los datos. Los métodos pueden ser: GET para obtener datos, POST para crear un objeto, PUT y PATCH para actualizar datos y DELETE para borrar. Para el intercambio de datos entre el cliente y el servidor se hizo uso de la notación JSON.

La Figura 2.18 presenta la manera en que los componentes de la herramienta se comunican.

Tabla 2.6 Lenguajes de programación y herramientas usadas para la implementación

Componente	Lenguaje de programación	Versión	Framework	Versión	IDE ²⁹ o editor de código
Backend	Python	3.7	Flask	1.1.2	Pycharm (IDE)
Frontend anotación	TypeScript	3.5.3	Angular	8.2.7	Visual Studio Code (Editor de código)
Frontend visualización	TypeScript	3.5.3	Angular	8.2.14*	Visual Studio Code (Editor de código)

*Se utilizó una versión superior de TypeScript ya que al momento de crear el proyecto esta se encontraba disponible. El componente de *frontend* de visualización se inició una vez concluido el componente *frontend* de anotación.

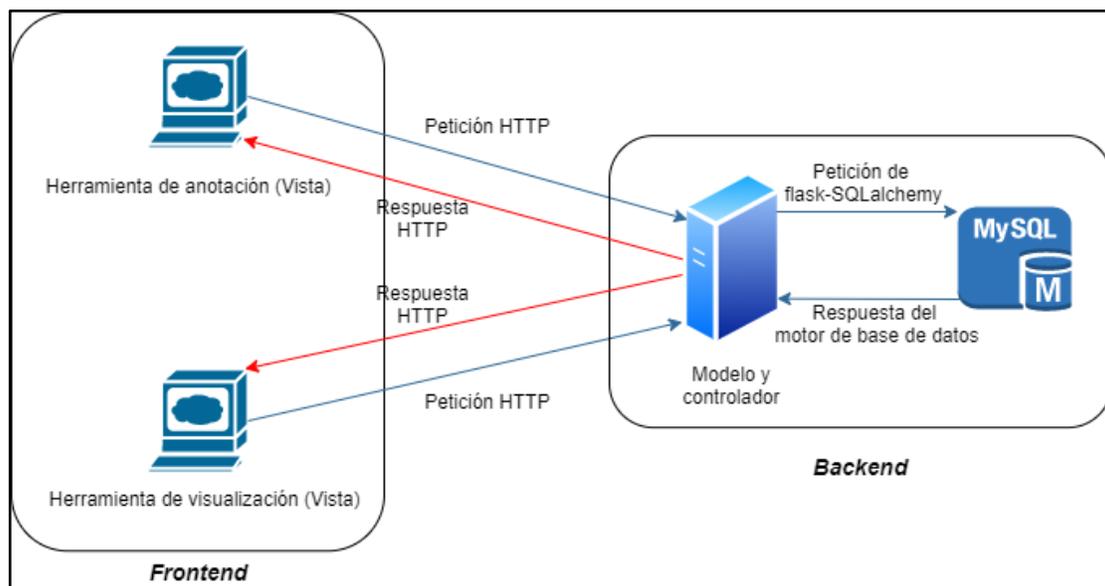


Figura 2.18 Comunicación de componentes de la herramienta

La estructura del código generado siguiendo la arquitectura MVC se presenta en la Figura 2.19. El código se divide en dos secciones principales:

²⁹ IDE: Un entorno de desarrollo integrado, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software



Figura 2.19 Estructura del código de la herramienta

- *Backend*: el modelo y el controlador se encuentran en esta sección de código. Para cada uno de estos componentes existe un directorio. Existen dos directorios adicionales *services* y *utils*. El directorio *services* almacena un archivo por cada modelo de datos, por ejemplo: para el modelo *usuario* corresponde el archivo *usuario_service*. Estos archivos contienen todas las operaciones relacionadas al acceso de datos de cada uno de los modelos. Por otro lado, el directorio *utils* tiene funcionalidades adicionales que se usan en la herramienta, por ejemplo: el envío de correo cuando se crea un usuario.
- *Frontend*: Angular crea el proyecto con ciertos directorios y archivos. Dentro del directorio *app* se encuentra el código de los componentes que forman la aplicación. Por ejemplo: el directorio *paginas* contiene componentes como anotación, consolidación, administración de usuarios, entre otros.

A continuación se explican las secciones de código más importantes generadas en cada *sprint*.

2.4.1. MÓDULO DE ADMINISTRACIÓN - *SPRINT 1*

En esta sección se presenta el código relacionado a la implementación del *sprint*: Módulo de administración. Este *sprint* incluye la implementación de: arranque de la aplicación, inicio y cierre de sesión, y administración de los elementos: usuarios, políticas, tratamientos, atributos y valores. También incluye el código en el *backend* para la administración de los elementos: rol de usuario, módulos y colores. Estos elementos no fueron incluidos en la interfaz gráfica ya que en conjunto con el *product owner* se decidió que son elementos con los cuales no se interactúa de manera constante y por lo tanto no

necesitan una interfaz gráfica, la administración de estos elementos se hará a través de peticiones a las APIs RESTful correspondientes. El arranque de la aplicación de *backend* y *frontend* se puede encontrar en el Anexo C y Anexo D respectivamente.

2.4.1.1. Administración de elementos

Una vez inicializadas las aplicaciones de *backend* y *frontend* es posible continuar con la programación de la administración de los distintos elementos que conforman la herramienta. Como ejemplo, se muestra el elemento `Usuario`, todos los demás elementos (política, tratamiento, atributo y valor) siguen una lógica similar en su código y por ello no se considera necesaria su explicación.

El Código 2.2 presenta la clase `Usuario`. Esta clase es un modelo de datos ya que los atributos de la clase corresponden a columnas de la tabla y una instancia de la clase corresponde a una fila. Como se observa en el código la clase `Usuario` hereda de la clase `Model`, esta clase declara un modelo de *SQLAlchemy*. La línea 10 declara el nombre de la tabla en la base de datos. De la línea 12 a la 19 se declaran los atributos de la clase. La línea 15 permite relacionar a la clase `Usuario` con `RolUsuario` donde se declara a `RolUsuario` como clave foránea de la clase `Usuario`.

```
8 class Usuario(db.Model):
9     """Modelo de Usuario"""
10    __tablename__ = "usuario"
11
12    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
13    email = db.Column(db.String(255), unique=True, nullable=False)
14    hora_registro = db.Column(db.DateTime, nullable=False)
15    rol_usuario = db.Column(db.Integer, db.ForeignKey('rol_usuario.id'))
16    clave_hash = db.Column(db.String(100))
17    activo = db.Column(db.Boolean)
18    entrenamiento = db.Column(db.Boolean)
19    anotaciones = db.relationship("Anotacion", backref=db.backref("usuario"))
20
```

Código 2.2 Clase Usuario

Una vez codificada la clase y sus métodos es posible programar los servicios de la clase usuario. Los servicios se encargan de la comunicación entre el controlador y el modelo de datos, y realizan acciones de escritura y lectura con la base de datos. Las acciones de CRUD³⁰ son realizadas en esta sección de código.

³⁰ **CRUD** (por sus siglas en inglés): Crear, leer, actualizar y eliminar son las acciones que se pueden ejecutar sobre un modelo de datos.

En el Código 2.3 se presenta la función para guardar un usuario. La función recibe como parámetro un objeto `usuario`. En la línea 22 de código se consulta un usuario mediante su email, luego, en el condicional en caso de no existir un usuario registrado con este email se procede a crear uno nuevo, caso contrario se retorna un mensaje de error. En la línea 24 se genera una clave aleatoria que es adjuntada al correo que se envía al usuario con sus credenciales. En la línea 25 se comprueba que el envío de correo haya sido exitoso para luego crear un `nuevo_usuario` que es una instancia de la clase `Usuario`, es decir una nueva fila de la tabla. Para persistir los cambios se llama a la función `guardar_cambios` y se envía como parámetro a `nuevo_usuario`. Finalmente, una vez guardado el usuario se envía un mensaje de respuesta de éxito.

```
21 def guardar_nuevo_usuario(usuario):
22     user = Usuario.query.filter_by(email=usuario['email']).first()
23     if not user:
24         clave = clave_aleatoria()
25         if enviar_correo(usuario['email'], clave, usuario['rol_usuario']):
26             nuevo_usuario = Usuario(
27                 email=usuario['email'],
28                 hora_registro=datetime.datetime.now(),
29                 rol_usuario=usuario['rol_usuario'],
30                 clave=clave,
31                 activo=True,
32                 entrenamiento=usuario['entrenamiento']
33             )
34             guardar_cambios(nuevo_usuario)
35             mensaje = respuesta(True, 'Usuario registrado exitosamente')
36             return mensaje, 201
37         else:
38             mensaje = respuesta(False, 'Error al enviar correo. Pongase en contacto con soporte')
39             return mensaje, 409
40     else:
41         mensaje = respuesta(False, 'El usuario ya existe, por favor inicie sesion')
42         return mensaje, 409
43
```

Código 2.3 Función Guardar Usuario

La función `respuesta` invocada en la línea 35 del código anterior sirve para crear mensajes de respuesta, esta función se presenta en el Código 2.4.

```
1 def respuesta(exito, mensaje):
2     estado = 'exito' if éxito else 'fracaso'
3
4     respuesta = {
5         'estado': estado,
6         'mensaje': mensaje
7     }
8
9     return respuesta
10
```

Código 2.4 Función respuesta

Lo presentado en el Código 2.3 corresponde a la acción de crear, ahora se presenta el código necesario para obtener un objeto desde la base de datos. Un usuario y en general cualquier modelo de datos puede ser obtenido por cualquier de sus atributos o conjunto de atributos que lo caracteriza, es decir que lo haga único. En el Código 2.5 se presenta la función para obtener un usuario mediante su `id`.

```
90 def obtener_usuario(id):
91     return Usuario.query.filter_by(id=id).first()
92
```

Código 2.5 Función obtener Usuario

En el Código 2.6 se presenta el código para la edición de usuario. Primero, se verifica que el usuario exista en la base de datos, una vez comprobado se procede a editar los campos del usuario y guardar estos cambios en la base de datos, finalmente se retorna un mensaje de éxito. En caso de no existir el usuario a editar se retorna un mensaje de error.

```
45 def editar_usuario(usuario):
46     usuario = Usuario.query.filter_by(id=usuario['id']).first()
47     if usuario:
48         usuario.email = usuario['email']
49         usuario.rol_usuario = usuario['rol_usuario_id']
50         usuario.activo = usuario['activo']
51         usuario.entrenamiento = usuario['entrenamiento']
52
53         guardar_cambios(usuario)
54         mensaje = respuesta(True, 'Usuario editado con éxito')
55         return mensaje, 201
56
57     else:
58         mensaje = respuesta(False, 'No existe el usuario')
59         return mensaje, 409
60
```

Código 2.6 Función editar Usuario

En el Código 2.7 se presenta la función `eliminar_usuario`. Esta sección de código se encierra en un bloque `try...except`. En caso de existir conflictos en la base de datos o que exista algún otro elemento que dependa de este registro (integridad referencial) se ejecutará el código del bloque `except`, este bloque deshace los cambios que pudieron realizarse en la base de datos y retorna un mensaje de error. En caso de eliminar el usuario con éxito se retorna un mensaje con esta información, esto se realiza en el bloque `else`.

```

62 def eliminar_usuario(id):
63     try:
64         Usuario.query.filter_by(id=id).delete()
65     except:
66         db.session.rollback()
67         mensaje = respuesta(False, 'Error eliminando usuario')
68         return mensaje, 409
69     else:
70         db.session.commit()
71         mensaje = respuesta(True, 'Usuario eliminado con éxito')
72         return mensaje, 201
73

```

Código 2.7 Función eliminar Usuario

En el Código 2.8 se presenta el código correspondiente al controlador del modelo Usuario, el cual contiene los *endpoints* creados para realizar las acciones de CRUD. Se hace uso del decorador `@api.route` (línea 14), el cual recibe como parámetros:

- Un *string* que define la ruta del recurso, esta ruta será concatenada con un *path* definido en el Código 2.9.
- Una lista con los métodos HTTP que están permitidos en esa ruta determinada.

La clase `Usuario` es una subclase de `Resource` provista por `Flask-restplus`. La documentación oficial [30] recomienda que un recurso extienda esta clase y exponga los métodos para cada método HTTP soportado por la clase. En caso de que un recurso sea solicitado con un método HTTP no soportado se obtiene una respuesta con estado HTTP 405: “Método no Permitido”, caso contrario el método adecuado es ejecutado.

```

14 @api.route('/', methods=['GET', 'POST', 'PATCH'])
15 class Usuario(Resource):
16     @api.doc('lista de usuarios registrados')
17     @api.marshal_list_with(_usuarioConsultar)
18     def get(self):
19         return obtener_usuarios()
20
21     @api.response(201, 'Usuario registrado exitosamente')
22     @api.doc('Crear un nuevo usuario')
23     @api.expect(_usuario, validate=True)
24     def post(self):
25         data = request.json
26         return guardar_nuevo_usuario(usuario=data)
27
28     @api.response(201, 'Usuario editado exitosamente')
29     @api.doc('Editar un usuario')
30     @api.expect(_usuarioEditar, validate=True)
31     def patch(self):
32         data = request.json
33         return editar_usuario(usuario=data)
34

```

Código 2.8 *Endpoint* para CRUD de Usuario

En la línea 8 del Código 2.8 la variable `api` es una instancia `UsuarioDto.api`. Esta a su vez es una instancia de la clase `Namespace` provista por `flask-restplus`. Esta clase provee al controlador algunos decoradores, por ejemplo `@api.route`. Algunos de los decoradores que `Namespace` provee son:

- `api.route`: es usado para definir las rutas de un recurso.
- `api.marshal_with`: es usado para especificar los atributos con los que se debe serializar la respuesta de ese servicio.
- `api.marshal_list_with`: es usado para realizar la serialización de listas. Por ejemplo, en la línea 17 del Código 2.8 se hace uso de este decorador para realizar el *marshalling* de una lista de `Usuarios`.
- `api.doc`: es un decorador que sirve para añadir un poco de documentación al objeto en el que se lo usa.
- `api.response`: es un decorador para especificar una de las respuestas del recurso.
- `api.expect`: es un decorador para especificar el modelo que se espera. Este decorador es especialmente útil para verificar que todos los parámetros esperados lleguen desde las peticiones de *frontend*.
- `api.param`: es un decorador que permite especificar uno de los parámetros esperados.

En el Código 2.9 se declara las rutas principales de cada controlador, estos objetos contienen toda la información referente a los servicios web que expondrá el *backend*.

```
25 api.add_namespace(auth_ns)
26 api.add_namespace(anoacion_ns, path='/Anotacion')
27 api.add_namespace(atributo_ns, path='/Atributo')
28 api.add_namespace(color_ns, path='/Color')
29 api.add_namespace(modulo_ns, path='/Modulo')
30 api.add_namespace(politica_ns, path='/Politica')
31 api.add_namespace(rol_usuario_ns, path='/RolUsuario')
32 api.add_namespace(tratamiento_ns, path='/Tratamiento')
33 api.add_namespace(usuario_ns, path='/Usuario')
34 api.add_namespace(valor_ns, path='/Valor')
```

Código 2.9 Registro de *paths* de *endpoints* del *backend*

Por otro lado, en el Código 2.10 presenta la declaración de la clase `UsuarioDto` la cual contiene un objeto `api` y objetos que a su vez son instancia de `api.model`. Estos objetos son útiles para realizar la serialización de los objetos generados con Python a un objeto JSON, que luego será recibido e interpretado por el *frontend* de la herramienta.

```
244 class UsuarioDto:
245     api = Namespace('Usuario', description='Operaciones relacionadas a usuarios ')
246
247     usuario = api.model('usuario', {
248         'email': fields.String(required=True, description='direccion de email de usuario/nombre de usuario'),
249         'rol_usuario_id': fields.Integer(required=True, description='rol de usuario'),
250         'entrenamiento': fields.Boolean(required=True, description='el usuario sera sometido a entrenamiento?')
251     })
252
253     usuarioConsultar = api.model('usuarioConsultar', {
254         'id': fields.Integer,
255         'email': fields.String,
256         'rol_usuario_id': fields.Integer,
257         'rol_usuario': fields.String,
258         'activo': fields.Boolean,
259         'entrenamiento': fields.Boolean
260     })
```

Código 2.10 Clase `UsuarioDTO`

Todo el código presentado hasta ahora corresponde al *backend* para la administración de `Usuario`. Ahora, se presenta el código de *frontend* para la administración de este elemento.

En la Figura 2.20 se presentan los archivos que forman el componente `Usuario`. También se puede encontrar el componente `usuario-dialogo`, el cual es utilizado como modal para acciones de creación y edición de usuarios.

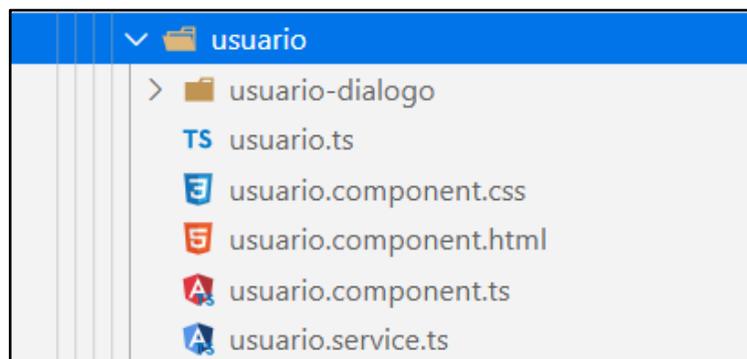


Figura 2.20 Componente `Usuario`

El archivo `usuario.component.ts` se presenta en los códigos: Código 2.11, Código 2.12, Código 2.13 y Código 2.14. A continuación, se describe cada uno de estos.

En el Código 2.11 se presentan las importaciones necesarias para el componente. Luego de la línea 8 a la 12 se encuentra el decorador `@Component` el cual recibe un diccionario que contiene:

- `selector`: define el selector para que pueda ser usado por otros componentes.
- `templateUrl`: archivo que contiene el código HTML.
- `styleUrls`: archivo que contiene el código CSS.

```
1 import { Component } from '@angular/core';
2 import { UsuarioConsultar } from './usuario'
3 import { UsuarioService } from './usuario.service'
4 import { MatDialog, MatSnackBar, MatSnackBarConfig, MatTableDataSource } from '@angular/material';
5 import { UsuarioDialogoComponent } from './usuario-dialogo/usuario-dialogo.component';
6 import { NotificacionComponent } from 'src/app/notificacion/notificacion.component';
7
8 @Component({
9   selector: 'app-usuario',
10  templateUrl: './usuario.component.html',
11  styleUrls: ['./usuario.component.css']
12 })
```

Código 2.11 Componente usuario 1 de 4

Así también, en el Código 2.12 se presenta la clase `UsuarioComponent`. Esta sección de código contiene la estructura básica con la que se construyeron los componentes en esta herramienta. Un componente está dividido en tres secciones:

- Variables: de la línea 15 a la 25 se definen las variables necesarias para el componente.
- Constructor: de la línea 27 a la 32 se define el constructor de la clase que se ejecuta cada vez que se realiza una instancia de esta clase. Este es el primer método en ejecutarse.
- Funciones: de la 32 en adelante (Código 2.13 y Código 2.14) se presentan las funciones definidas para este componente.

Esta estructura permite tener un código mejor organizado y será seguida por todos los componentes que forman la herramienta.

En el Código 2.13 se presenta el código que permite abrir al componente `UsuarioDialogoComponent` como un diálogo y establecer ciertas propiedades como el ancho (`width`) y el alto (`height`) que tendrá el modal. El parámetro `data` permite enviar datos requeridos entre el componente padre y el modal.

```

13 export class UsuarioComponent {
14
15     dataSource: MatTableDataSource<UsuarioConsultar>;
16     displayedColumns = ['id', 'email', 'rol_usuario', 'activo', 'entrenamiento', 'editar', 'eliminar'];
17     usuarioNuevoAux: UsuarioConsultar = {
18         id: 0,
19         email: '',
20         rol_usuario_id: 2,
21         rol_usuario: '',
22         entrenamiento: true,
23         activo: true
24     }
25 }
26
27 constructor(
28     private _usuarioService: UsuarioService,
29     private _dialogo : MatDialog,
30     private _notificacion : MatSnackBar,
31
32 ) { this.consultarUsuarios()}

```

Código 2.12 Componente usuario 2 de 4

```

34     nuevoUsuario() {
35         const dialogoNuevo = this._dialogo.open(UsuarioDialogoComponent, {
36             width: '40%',
37             height: 'fit-content',
38             data: {
39                 usuario: this.usuarioNuevoAux,
40                 nuevo: true
41             }
42         })
43
44         dialogoNuevo.afterClosed().subscribe(
45             () => this.consultarUsuarios()
46         )
47     }

```

Código 2.13 Componente usuario 3 de 4

Finalmente, el Código 2.14 contiene la función `consultarUsuario` que a su vez hace uso de la función definida en el servicio del componente `Usuario` (`usuario.service.ts`) presentada en el Código 2.15. Esta función se suscribe a la función `obtenerUsuarios` y en caso de éxito fija el valor obtenido a la variable `dataSource`, en caso de fracaso despliega un mensaje de error en la pantalla.

```

78     consultarUsuarios() {
79         this._usuarioService.obtenerUsuarios().subscribe(
80             resultado => this.dataSource = new MatTableDataSource(resultado),
81             error => {
82                 this.notificacion("No ha sido posible obtener los usuarios", "fracaso-snackbar", 4000);
83             }
84         )
85     }
86

```

Código 2.14 Componente Usuario 4 de 4

El Código 2.15 presenta parte del código del archivo `usuario.service.ts`. Este código permite conectar a Angular con el *backend* de la herramienta. Con la ayuda de la librería `http` es posible consumir los servicios expuestos por el *backend* a través de una petición HTTP. De la línea 16 a la 18 se observa la función `obtenerUsuarios` que tiene un tipo de retorno `Observable<UsuarioConsultar[]>`. Un `Observable` es una clase provista por la librería `rxjs` y es una representación de un conjunto cualquiera de valores en cualquier periodo de tiempo [31]. Un `Observable` implementa el patrón de diseño publicador/suscriptor³¹, por esto la función `consultarUsuarios` (Código 2.14) realiza una suscripción de la función `obtenerUsuarios`.

```
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4  import { UsuarioConsultar, UsuarioAsignar } from './usuario'
5  import { environment } from 'src/environments/environment';
6  import { Respuesta } from 'src/app/tipos';
7
8  @Injectable({
9    providedIn: 'root'
10 })
11
12 export class UsuarioService {
13   url = environment.url + 'Usuario/';
14   constructor(private http: HttpClient) {}
15
16   obtenerUsuarios(): Observable<UsuarioConsultar[]>{
17     return this.http.get<UsuarioConsultar[]>(this.url);
18   }

```

Código 2.15 Servicio de Usuario

El Código 2.16 presenta la interfaz `UsuarioConsultar`. Una interfaz es un contrato que define los atributos y métodos que un objeto debe implementar. En este caso esta interfaz es usada para deserializar los objetos que recibe la aplicación al consultar los usuarios desde el *backend* de la aplicación.

```
16  export interface UsuarioConsultar {
17    id: number,
18    email: string,
19    rol_usuario_id: number,
20    rol_usuario: string,
21    activo: boolean,
22    entrenamiento: boolean,
23  }

```

Código 2.16 Interfaz `UsuarioConsultar`

³¹ **Publicador/subscriptor:** permite anunciar eventos a múltiples consumidores, llamados suscriptores, interesados de forma asincrónica, sin acoplar los remitentes a los receptores.

Por otro lado, en el Código 2.17 y Código 2.18 se presenta el código HTML usado para el listado de usuarios en la herramienta. El Código 2.17 es usado para presentar el encabezado del listado de usuarios. En la línea 8 se observa que el evento `click` del botón Nuevo es enlazado con la función `nuevoUsuario`, esta acción permite abrir el modal para crear un nuevo usuario.

```

<mat-card class="mat-elevation-z8 card" style="background-color: #fafafa ;">
  <mat-grid-list cols="4" rowHeight="40px">
    <mat-grid-tile [colspan]="3" [rowspan]="1">
      <mat-card-title style="width: 100%;">Gestión de Usuarios</mat-card-title>
    </mat-grid-tile>
    <mat-grid-tile [colspan]="1" [rowspan]="2">
      <button mat-raised-button class="standar" style="height: 35px; color: white"
        (click)="nuevoUsuario()">+Nuevo</button>
    </mat-grid-tile>
    <mat-grid-tile [colspan]="3" [rowspan]="1">
      <mat-card-subtitle style="width: 100%;">
        Lista, creación, eliminación y edición de usuarios
      </mat-card-subtitle>
    </mat-grid-tile>
  </mat-grid-list>

```

Código 2.17 Código HTML de interfaz de usuario 1/2

De igual manera, en el Código 2.18 se presenta el código que genera la interfaz gráfica para filtrar usuarios y para listar los usuarios obtenidos del `backend`. El elemento `table`, línea 21, se enlaza al valor de la variable `dataSource` del componente. Luego cada columna de la tabla se define con `ng-container` (ej: línea 25) que contiene el nombre de la columna y el valor de cada una de las filas.

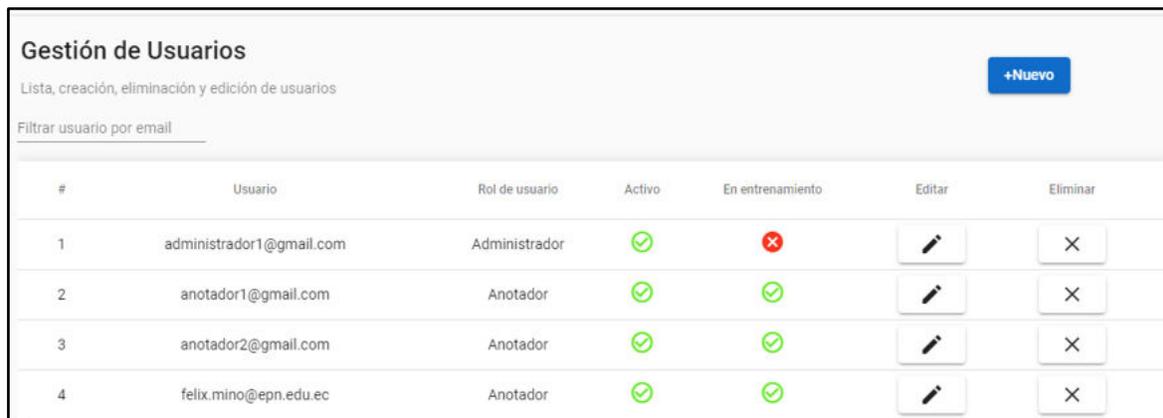
```

15 <mat-card-content style="height: 450px; overflow-x: hidden; overflow-y: auto;">
16
17 <mat-form-field>
18   <input matInput (keyup)="aplicarFiltro($event.target.value)" placeholder="Filtrar usuario por email">
19 </mat-form-field>
20
21 <table mat-table [dataSource]="dataSource" class="mat-elevation-z8" style="width: 100%;">
22   <tr mat-header-row *matHeaderRowDef="displayedColumns; sticky: true"></tr>
23   <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
24
25   <ng-container matColumnDef="id">
26     <th mat-header-cell *matHeaderCellDef"> # </th>
27     <td mat-cell *matCellDef="let usuario; let indice = index"> {{indice+1}}</td>
28   </ng-container>
29
30   <ng-container matColumnDef="email">
31     <th mat-header-cell *matHeaderCellDef"> Usuario</th>
32     <td mat-cell *matCellDef="let usuario"> {{usuario.email}}</td>
33   </ng-container>
34
35   <ng-container matColumnDef="rol_usuario">
36     <th mat-header-cell *matHeaderCellDef"> Rol de usuario</th>
37     <td mat-cell *matCellDef="let usuario"> {{usuario.rol_usuario}}</td>
38   </ng-container>
39

```

Código 2.18 Código HTML de interfaz de usuario 2/2

Finalmente, el resultado de Código 2.17 y Código 2.18 se presenta en la Figura 2.21.



#	Usuario	Rol de usuario	Activo	En entrenamiento	Editar	Eliminar
1	administrador1@gmail.com	Administrador	✓	✗		
2	anotador1@gmail.com	Anotador	✓	✓		
3	anotador2@gmail.com	Anotador	✓	✓		
4	felix.mino@epn.edu.ec	Anotador	✓	✓		

Figura 2.21 Resultado de: Código 2.17 y Código 2.18

En resumen, en esta sección se presentó la estructura y el código base para arrancar la aplicación tanto de *frontend* como de *backend* (anexos), el código necesario para construir la API RESTful de administración de usuarios y finalmente se presentó el código de Angular necesario para mostrar el listado de usuarios que retorna el *backend*.

2.4.2. MÓDULO DE ANOTACIÓN - *SPRINT* 2

En esta sección se presenta el código relevante de la implementación del *sprint*: módulo de administración. Este *sprint* incluye la implementación de: visualización (listado) de políticas de privacidad por anotar, anotación de una política de privacidad, visualización, edición y eliminación de anotaciones y notificación de inconsistencia al anotar.

El código correspondiente al listado de políticas de privacidad por anotar no se presenta por su similitud con el código explicado en la sección previa en cuanto al listado de usuarios.

2.4.2.1. Anotación de una política de privacidad

La página para la anotación de una política de privacidad está formada por tres componentes, como se evidencia en la Figura 2.22:

- **Componente padre:** este componente es un contenedor de los otros dos componentes. El nombre de este componente es `anotacion-politica`.
- **Componente para selección de texto:** este componente tiene los elementos y el código necesario para capturar la selección del texto del recuadro que contiene la sección de la política de privacidad. El nombre de este componente es `seleccionar-texto`.

- **Componente para selección de tratamientos de datos:** este componente presenta al usuario los tratamientos de datos con sus atributos y valores. Al menos un valor debe ser seleccionado para poder realizar una anotación. El nombre de este componente es `tree-view-tratamientos`.

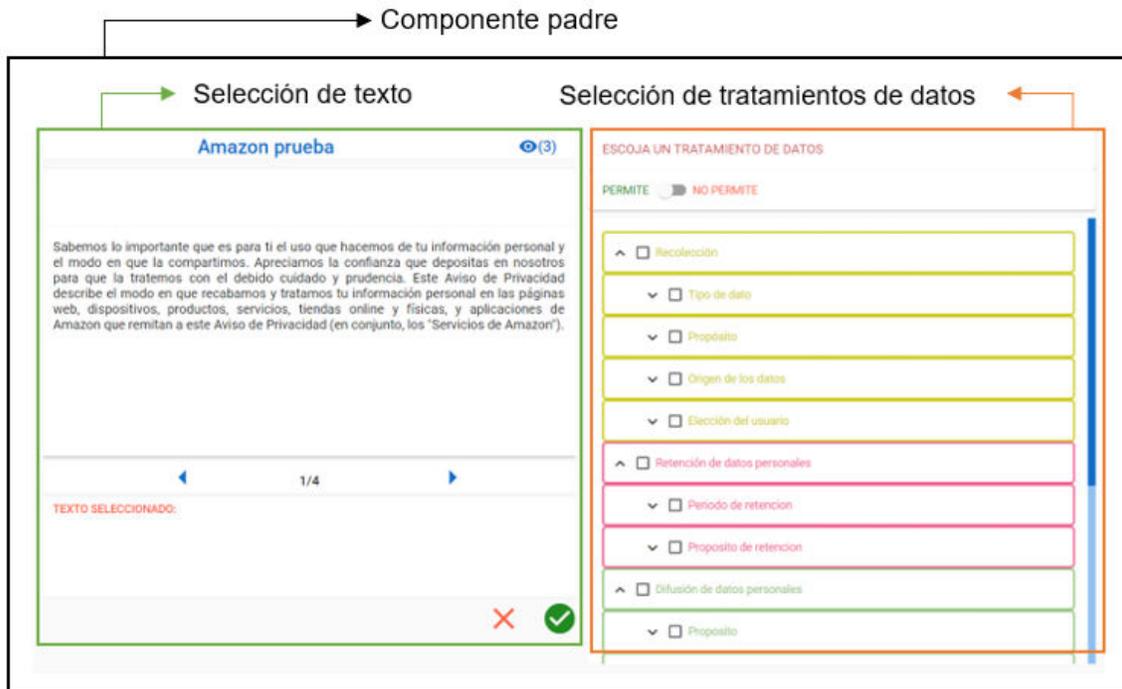


Figura 2.22 Componentes de la página de anotación

Ya que estos componentes necesitan comunicarse entre ellos, por ejemplo, el componente `anotacion_politica` recibe: 1) el texto seleccionado desde el componente `seleccionar_texto` y 2) los valores escogidos del componente `tree_view_tratamientos`. Esta comunicación entre componentes se realiza mediante un `servicio` haciendo uso de la clase `BehaviorSubject` provista por la librería `rxjs`. `BehaviorSubject` es una variante de `Subject`³² que requiere un valor inicial y emite su valor actual cuando un consumidor se suscribe a él [32].

En el Código 2.19 se presenta el servicio que permite comunicar (obtener y modificar valores) entre el componente padre (`anotacion-politica`) y el componente (`seleccionar-texto`). Las variables `texto` y `textoHtml` son una instancia de `BehaviorSubject` y a través de las funciones `obtenerTexto`, `obtenerTextoHtml`, `colocarTexto` y `colocarTextoHtml` pueden ser modificadas y leídas por los

³² **Subject:** es un tipo especial de `Observable` que permite la difusión múltiple de valores a varios observadores, al contrario de un `Observable` que permite difusión simple.

componentes que así lo requieran. La función `colocarTextoHtml` es útil para mantener una representación visual del texto seleccionado en todo momento.

```
10 export class SelectTextBoxService {
11
12     private texto = new BehaviorSubject<string>("");
13     private textoHtml = new BehaviorSubject<string>("");
14
15     private numeroAnotacionesParrafo = new BehaviorSubject<number>(0);
16
17     constructor(
18         private _anotacionService : AnotacionService,
19         @Inject(DOCUMENT) private documento: Document,
20     ) { }
21
22     public obtenerTexto(){
23         return this.texto
24     }
25
26     public obtenerTextoHmtl(){
27         return this.textoHtml
28     }
29
30     public colocarTexto(texto : string){
31         this.texto.next(texto)
32     }
33
34     public colocarTextoHtml(textoHtml : string){
35         this.textoHtml.next(textoHtml)
36         let input = this.documento.getElementById("seleccion");
37         input.innerHTML = textoHtml
38     }
39 }
```

Código 2.19 Servicio para comunicación de valores entre componentes

Por otro lado, en el Código 2.20 se presenta el código del componente `anotacion-politica` el cual se suscribe a las instancias de `BehaviorSubject` (`texto` y `textoHtml`) presentadas en el código anterior. De esta manera en cada actualización de estas instancias este componente recibirá el nuevo valor.

```
40 constructor(
41     private _router: Router,
42     private _anotacionService: AnotacionService,
43     private _seleccionarTextoService: SelectTextBoxService,
44     private _treeViewService: TreeViewCheckService,
45     private _usuarioService: UsuarioService,
46     private _dialogo: MatDialog,
47     private _notificacion: MatSnackBar
48 ) {
49     this.politicaId = this._router.getCurrentNavigation().extras.state.politica_id;
50
51     this._seleccionarTextoService.obtenerTexto().subscribe(
52         texto => this.texto = texto
53     )
54     this._seleccionarTextoService.obtenerTextoHmtl().subscribe(
55         textoHtml => this.textoHtml = textoHtml
56     )
57
58     this._treeViewService.obtenerPermite().subscribe(
59         permite => this.permite = permite
60     )
61 }
```

Código 2.20 Ejemplo de suscripción a un `BehaviorSubject`

Complementando lo anterior, en el Código 2.21 se presenta el código para obtener el texto seleccionado de la política de privacidad. El texto será capturado cuando el cursor se suelte sobre el cuadro que contiene la sección de la política de privacidad.

```
91  seleccion() {
92
93      this.limpiarTextoEscogido()
94
95      let seleccion = this.documento.getSelection();
96
97      seleccion.getRangeAt(0).cloneContents().childNodes.forEach(
98          item => {
99              //Se verifica si existen nodos del tipo BR
100             //si se encuentra alguno se reemplaza por la etiqueta html <br>
101             //caso contrario no se modifica
102             if (item.nodeName == "BR") {
103                 this.textoSeleccionadoHtml += "<br>";
104                 this.textoSeleccionado += " ";
105             }
106             else {
107                 this.textoSeleccionadoHtml += item.textContent;
108                 this.textoSeleccionado += item.textContent;
109             }
110         }
111     )
112
113     //Se verifica que la selección contenga texto
114     if (this.textoSeleccionado != "") {
115         this._seleccionarTextoService.colocarTexto(this.textoSeleccionado)
116         this._seleccionarTextoService.colocarTextoHtml(this.textoSeleccionadoHtml)
117     }
118 }
119
```

Código 2.21 Función para selección de texto

2.4.2.2. Notificación de inconsistencia en anotación

Finalmente, en este *sprint* se programó la implementación de notificación de inconsistencias para usuarios en entrenamiento. Esta característica permite a los usuarios que están iniciando en la anotación de políticas de privacidad tener sugerencias de tratamientos de datos para anotar sobre una sección específica. En la Figura 2.13 se muestra el diagrama de flujo que representa ese proceso, este diagrama sintetiza tanto el código de *frontend* como de *backend*. El código implementado para la notificación de inconsistencias se encuentra en el ANEXO F.

2.4.3. MÓDULO DE CONSOLIDACIÓN - *SPRINT 3*

En este *sprint* se implementó las siguientes funcionalidades: cálculo de coeficiente interanotador, consolidación de una política de privacidad, visualización, edición y eliminación de anotaciones en consolidación, visualización y copia de anotaciones de la etapa de anotación en consolidación y notificación de sección inconsistente en consolidación. Se

explicará el código correspondiente al cálculo de coeficiente inter-anotador y la notificación de sección inconsistente en consolidación, el resto de las funcionalidades no serán explicadas, ya que corresponden a componentes presentados en secciones previas o su código es muy similar.

2.4.3.1. Cálculo de coeficiente inter-anotador

Para el cálculo del coeficiente inter-anotador se hizo uso del coeficiente Alpha de Krippendorff y el cálculo se realizó con la ayuda de la librería *krippendorff*³³ implementada en Python. A continuación, se presenta una breve introducción a los conceptos básicos de Alpha de Krippendorff, para luego explicar la implementación en código.

Alpha de Krippendorff (α) es un coeficiente de confiabilidad desarrollado para medir el grado de concordancia entre observadores, codificadores, jueces, evaluadores o instrumentos de medición a través de la asignación de valores computables a fenómenos no estructurados. α nació en el análisis de contenido, pero es ampliamente usado en otros campos donde dos o más métodos para generar datos son aplicados a un mismo grupo de objetos, unidades de análisis o artículos y donde es necesario conocer el nivel de confiabilidad que presentan los resultados como para representar algo real [33]. La forma general de α se presenta en la ecuación 2.1.

$$\alpha = 1 - \frac{D_o}{D_e} \quad (2.1)$$

Donde D_o es la discrepancia observada entre los anotadores sobre las unidades de análisis, se presenta en la ecuación 2.2.

$$D_o = \frac{1}{n} \sum_c \sum_k O_{ck \text{ metric}} \delta_{ck}^2 \quad (2.2)$$

Y D_e es la discrepancia esperada cuando la codificación de las unidades se puede atribuir al azar, se presenta en la ecuación 2.3.

$$D_e = \frac{1}{n(n-1)} \sum_c \sum_k n_c * n_k \text{ metric} \delta_{ck}^2 \quad (2.3)$$

Algunos de los beneficios de usar el coeficiente de Alpha de Krippendorff frente a otros coeficientes son los siguientes [33]:

³³ <https://github.com/pln-fing-udelar/fast-krippendorff/>

- α funciona para cualquier número de observadores.
- α funciona para cualquier número de categorías o medidas.
- α funciona para cualquier métrica o nivel de medición (binario, nominal, ordinal, intervalo, razón, entre otros).
- α es capaz de trabajar con datos incompletos.
- α trabaja con muestras grandes o pequeñas, no requiere un mínimo de datos.

El cálculo del coeficiente se puede resumir en cuatro pasos:

1. Construcción de una matriz de datos de confiabilidad.
2. Construcción de la matriz de coincidencia en base a la matriz de datos.
3. Cálculo de las coincidencias observadas (O_{ck}) y de la función de diferenciación (δ^2_{ck}).
4. Cálculo del coeficiente mediante la ecuación de la forma general.

En cuanto a la interpretación del coeficiente, se presentan lo siguiente casos:

- $\alpha = 1$: los codificadores están perfectamente de acuerdo, lo que indica que los resultados son confiables.
- $\alpha = 0$: los codificadores están totalmente en desacuerdo y esto indica que no se debe confiar en estos resultados.
- $\alpha < 0$: un valor negativo indica que datos generados de manera aleatoria son más probables que los datos obtenidos y por lo tanto los datos no son fiables.

De acuerdo con Krippendorff, creador del coeficiente, un valor de $\alpha > 0.8$ es un valor aceptable y un límite para que los datos obtenidos sean considerados fiables, mientras que un valor en el rango de $0.667 < \alpha < 0.8$ puede ser considerado fiable solamente para elaborar conclusiones tentativas [28].

El cálculo de Alpha de Krippendorff en este proyecto se realizó para datos binarios sin datos faltantes. Es decir, para la forma más sencilla de datos de las expuestas en [33]. Para esta forma en particular el coeficiente calculado con la ecuación 2.1 se simplifica y puede ser calculado mediante la ecuación 2.4:

$$\alpha = 1 - (n - 1) \frac{O_{01}}{n_0 * n_1} \quad (2.4)$$

Donde los valores de O_{01} , n_0 , n_1 y n se obtienen de la matriz de coincidencia binaria, presentada en la Figura 2.23.

Valores:	0	1	
0	O_{00}	O_{01}	n_0
1	O_{10}	O_{11}	n_1
Total de valores:	n_0	n_1	$n = x * N$

Figura 2.23 Matriz de coincidencia binaria Alpha de Krippendorff

Donde x es el número total de observadores y N el número de columnas de la matriz de datos de confiabilidad.

Una matriz de datos de fiabilidad para datos binarios con tres observadores y diez unidades de análisis presenta la forma de la Tabla 2.7:

Tabla 2.7 Matriz de datos de confiabilidad binaria para tres codificadores y diez unidades de análisis

Unidades de análisis:	1	2	3	4	5	6	7	8	9	10
Anotador 1	0	1	1	0	0	0	1	1	0	1
Anotador 2	1	1	0	0	1	0	1	0	0	1
Anotador 3	0	1	0	0	0	1	1	1	0	1

Para este proyecto, los generadores de datos son los anotadores de una política de privacidad, mientras que los objetos de análisis son el conjunto de tratamiento/atributo/valor junto con su acción (ejecuta o no ejecuta) de las anotaciones de una política de privacidad. La decisión de realizar la comparación a nivel de tratamiento/atributo/valor se debe a que las políticas de privacidad suelen ser repetitivas, por lo que un mismo tratamiento/atributo/valor puede encontrarse en distintas secciones de una política de privacidad. Para el cálculo de este coeficiente basta con que un anotador realice la anotación (tratamiento/atributo/valor y acción) una vez a lo largo de la política de privacidad. Por lo tanto, los valores de la matriz serán llenados de acuerdo con el siguiente criterio:

- 0 (cero) si la unidad de análisis (tratamiento/atributo/valor y acción) no fue encontrada dentro de las anotaciones del anotador en específico.
- 0 (cero), se añade un cero en cada anotador por cada párrafo en el cual no existe ninguna anotación. Este caso, se considera como una coincidencia por parte de los anotadores, por lo tanto, se añade una columna de ceros en la matriz de datos de confiabilidad por cada sección no anotada.
- 1 (uno) si la unidad de análisis (tratamiento/atributo/valor y acción) existe en las anotaciones de un anotador en específico.

Este proceso es ilustrado en el diagrama de flujo presentado en la Figura 2.14. Y así mismo se presenta un ejemplo de cálculo de coeficiente en la herramienta en el ANEXO L.

En Código 2.22, Código 2.23 y Código 2.24 se presenta el código implementado para el cálculo del coeficiente inter-anotador. Cabe recordar que esta función es ejecutada cuando el último anotador asignado a una política de privacidad finaliza el proceso de anotación, es decir, luego de esto no existen cambios en la etapa de anotación.

```

366 def calcular_coeficiente_interanotador(politica_id, usuarios):
367     lista_anotaciones_usuarios = []
368
369     # Se consulta los valores que han sido anotados sobre la política por cada usuario
370     for usuario in usuarios:
371         valores = (db.session.query(Anotacion.ejecuta.distinct(), AnotacionValorRelacion.valor_id)
372                 .outerjoin(AnotacionValorRelacion, Anotacion.id == AnotacionValorRelacion.anotacion_id)
373                 .outerjoin(Parrafo, Anotacion.parrafo_id == Parrafo.id)
374                 .outerjoin(Politica, Parrafo.politica_id == Politica.id)
375                 .filter(Politica.id == politica_id,
376                       Anotacion.usuario_id == usuario[1].id,
377                       Anotacion.consolidar == False)
378                 .order_by(AnotacionValorRelacion.valor_id).all())
379
380     #Se crea una tupla (valor.id, anotacion.ejecuta) por cada valor anotado
381     anotaciones_usuario = []
382
383     for valor in valores:
384         tupla = (valor[0], valor[1])
385         anotaciones_usuario.append(tupla)
386
387     #Se adjunta a lista_anotaciones_usuarios los valores encontrados en este usuario.
388     #Una vez que se realiza el ciclo para todos los usuarios se obtendrán todos
389     #los valores que han sido anotados sobre la política
390     lista_anotaciones_usuarios.append(anotaciones_usuario)
391
392     #Esta variable contiene todos los valores anotados sobre la política
393     #Se eliminan tuplas repetidas
394     lista_unica_anotaciones = lista_anidada_unica(lista_anotaciones_usuarios)

```

```

396     #Se inicializa la matriz de datos de confiabilidad
397     matriz_datos_confiabilidad = []
398
399     #Se verifica que valores existen en cada usuario
400     #se añade un 1 en caso que el valor existe
401     #o un 0 en caso negativo.
402     for usuario in lista_annotaciones_usuarios:
403         datos_usuario = []
404         for valor in lista_unica_annotaciones:
405             if valor in usuario:
406                 datos_usuario.append(1)
407             else:
408                 datos_usuario.append(0)
409         matriz_datos_confiabilidad.append(datos_usuario)

```

Código 2.23 Cálculo de coeficiente inter-annotador 2 de 3

```

411     #Se consulta en la bdd cuantos parrafos de la politica contienen anotaciones
412     numero_parrafos_annotados = (db.session.query(Parrafo.id.distinct())
413     .outerjoin(Anotacion, Parrafo.id == Anotacion.parrafo_id)
414     .outerjoin(Politica, Parrafo.politica_id == Politica.id)
415     .filter(Anotacion.consolidar == False,
416     |         Politica.id == politica_id).count())
417
418     numero_parrafos_totales = consultar_num_parrafos_politica(politica_id)
419
420     #Se calcula el numero de párafos no anotados
421     numero_parrafos_no_annotados= numero_parrafos_totales - numero_parrafos_annotados
422
423     #Por cada parrafo no anotado se añade un 0
424     #en todos las filas de la matriz de datos de confiabilidad
425     if numero_parrafos_no_annotados == 0:
426         if all(x == matriz_datos_confiabilidad[0] for x in matriz_datos_confiabilidad):
427             for usuario in matriz_datos_confiabilidad:
428                 usuario.append(0)
429         else:
430             for i in range(0, numero_parrafos_no_annotados):
431                 for usuario in matriz_datos_confiabilidad:
432                     usuario.append(0)
433
434     #Se transforma la matriz de datos de confiabilidad a un array que la librería
435     #kirppendorff puede entender
436     datos_de_fiabilidad = array(matriz_datos_confiabilidad)
437
438     #Cálculo del coeficiente mediante la librería
439     return krippendorff.alpha(reliability_data=datos_de_fiabilidad, level_of_measurement='nominal')

```

Código 2.24 Cálculo de coeficiente inter-annotador 3 de 3

Este coeficiente fue implementado con el objetivo de ayudar al usuario que consolida una política de privacidad a conocer el grado de fiabilidad que puede tener respecto a las anotaciones realizadas en la etapa de anotación.

2.4.3.2. Notificación de sección inconsistente en consolidación

Por otro lado, en este *sprint* se implementó también la notificación de sección inconsistente en consolidación. Esta funcionalidad permite tener una advertencia visual de cada sección de una política de privacidad acerca de la coincidencia o no de las anotaciones realizadas

en la etapa de anotación por los distintos anotadores, es decir, si los anotadores realizaron las mismas anotaciones sobre una sección o no. El código implementado para esta funcionalidad se presenta en el ANEXO G.

2.4.4. MÓDULO DE VISUALIZACIÓN - *SPRINT* 4

Finalmente, en este *sprint* se implementaron las siguientes funcionalidades: página de bienvenida en herramienta de visualización, lista de políticas de privacidad para visualizar, visualización de una política de privacidad anotada y filtrado de anotaciones en visualización. Se explicará el código correspondiente a la visualización de una política de privacidad anotada y al filtrado de anotaciones en visualización.

Resulta importante aclarar que este *sprint* fue desarrollado sobre un nuevo proyecto de *frontend*. Como se indicó en secciones previas, el objetivo de este nuevo sitio es que sea accesible sin necesidad de la creación de una cuenta. Sin embargo, el *backend* de la aplicación es único y compartido, y es de donde se obtendrán los datos necesarios para la presentación de las políticas de privacidad.

2.4.4.1. Visualización de una política de privacidad

En cuanto a la visualización de una política de privacidad se usó de los colores asignados a cada uno de los tratamientos de datos para resaltar el texto de una anotación. En caso de que una anotación presente más de un valor y los valores pertenezcan a atributos de tratamientos distintos, se muestra el color del primer tratamiento bajo el cual se realizó la anotación. Cuando el usuario coloque el cursor sobre una anotación se presenta un cuadro informativo que contiene información relacionada con una anotación, concretamente se presenta: el número total de valores en la anotación, el tratamiento, atributo y valor de cada uno de los valores anotados, la acción (ejecuta o no ejecuta) y en caso de existir se presenta el comentario asociado a la anotación.

En esta interfaz el usuario también puede visualizar una lista de todos los tratamientos guardados en la herramienta (que contienen al menos un atributo y un valor), incluso si no existen anotaciones bajo este tratamiento. Cada uno de los tratamientos presenta una barra de color en el fondo que corresponde al porcentaje que representa este tratamiento del total de valores anotados, esto con el propósito de facilitar al usuario la lectura de la política de privacidad y mostrar de forma gráfica que acciones son las que prevalecen en una política de privacidad en particular. El *mockup* relacionado a esta funcionalidad se presenta en la Figura 2.11.

En la Figura 2.16 se presenta el diagrama de flujo del proceso que se ejecuta en el *frontend* de la aplicación para llevar a cabo la presentación de una política de privacidad. Para presentar una política de privacidad, primero es necesario obtener esta información desde el *backend* de la herramienta. Una vez obtenida la política de privacidad desde el *backend* se ejecuta la función `politicaOriginal` la cual presenta las anotaciones por defecto, es decir, sin aplicar ningún filtro sobre estas. El código de esta función se presenta a continuación en Código 2.25.

```

181 | //Presentación de política de privacidad sin filtros
182 | politicaOriginal() {
183 |   let politicaTexto: string = "";
184 |
185 |   this.presentacion.politica.parrafos.forEach(
186 |     parrafo => {
187 |       let parrafoCss: string = "";
188 |
189 |       if (parrafo.titulo != '') {
190 |         parrafoCss += this.darEstiloTitulo(parrafo);
191 |       }
192 |
193 |       parrafoCss += parrafo.texto_html;
194 |
195 |       parrafo.anotaciones.forEach(
196 |         anotacion => {
197 |           let textoCss: string =
198 |             this.darEstiloAnotacion(anotacion, anotacion.tratamientos[0].color_primario)
199 |
200 |           parrafoCss = parrafoCss.replace(anotacion.texto_html.trim(), textoCss)
201 |         }
202 |       )
203 |       politicaTexto += parrafoCss;
204 |     }
205 |   )
206 |   this.presentarPolitica(politicaTexto);
207 | }

```

Código 2.25 Función `politicaOriginal` (Presentación de política sin filtros)

Las funciones `darEstiloTitulo` y `darEstiloAnotacion` transforman texto plano en elementos HTML que contienen estilos CSS. En el Código 2.26 se presentan estas funciones respectivamente.

```

209 | darEstiloTitulo(parrafo: Parrafo):string {
210 |   return '<span style="font-weight: bold; font-size: 18px;">' + parrafo.titulo + '</span><br><br>'
211 | }
212 |
213 | darEstiloAnotacion(anotacion: Anotacion, color: string): string {
214 |   return ('<span class="anotacion" style="color: ' + color + '; cursor: pointer">'
215 |     + anotacion.texto_html + this.obtenerToolTip(anotacion) + "</span>")
216 | }
217 |

```

Código 2.26 Función `darEstiloTitulo` y `darEstiloAnotacion`

2.4.4.2. Filtrado de anotaciones en visualización

En esta sección se explica el funcionamiento de la característica de filtrado de anotaciones a nivel de tratamiento de datos y atributo.

Como se muestra en la Figura 2.24, el usuario tiene una lista de los tratamientos de datos que existen en la herramienta. Al seleccionar cualquiera de estos tratamientos se ejecuta el filtrado en base al tratamiento seleccionado y se despliega la lista de atributos que le pertenecen. Si el usuario selecciona uno de los atributos desplegados se realiza el filtrado por dicho atributo y se muestra un botón para limpiar todos los filtros presentes, esta interfaz se muestra en la Figura 2.25. Por el contrario, si el usuario selecciona otro tratamiento, el filtrado previo es borrado y se ejecuta un nuevo filtrado. Para limpiar un filtro a nivel de tratamiento es necesario volver a seleccionar el tratamiento con el cual fue ejecutado el filtro. Este comportamiento es resumido en el diagrama de flujo presentado en la Figura 2.17.

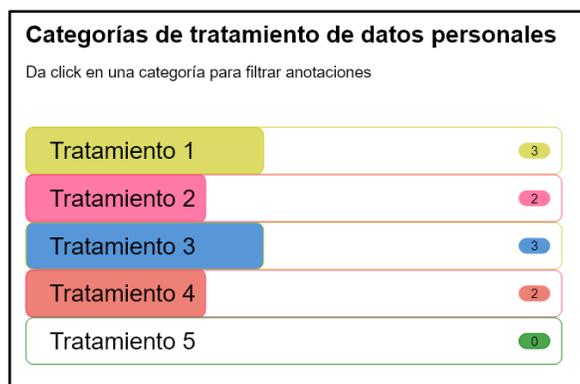


Figura 2.24 Lista de Tratamientos presentados al usuario para filtrado

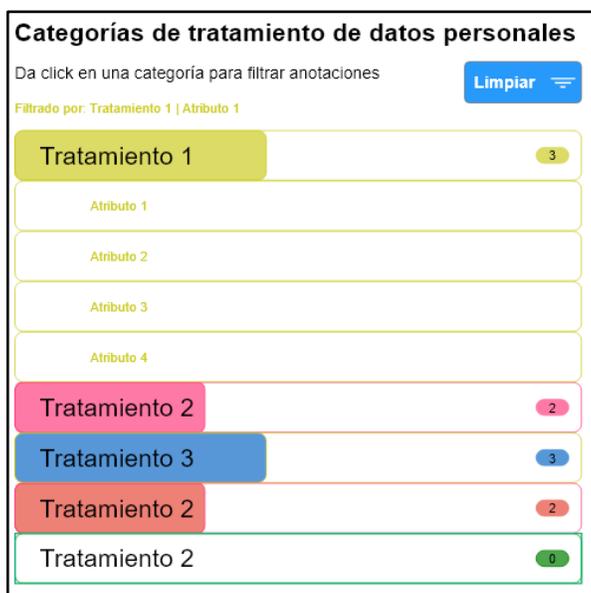


Figura 2.25 Interfaz de filtrado a nivel de atributo (opción de limpieza de filtros)

Parte del código implementado para esta funcionalidad se presenta en Código 2.27, Código 2.28 y Código 2.29. Este conjunto de código corresponde al filtrado de anotaciones por

tratamiento de datos, para el caso de filtrado por atributo no se presenta el código debido a su similitud.

Por un lado, el Código 2.27 es usado para manejar el estado de filtro de tratamiento, por ejemplo: contiene la información del filtro actual. Este código está enlazado con código HTML. El enlazado se presenta en la línea 41 del Código 2.28, cuando se da *click* sobre un tratamiento se ejecuta la función `manejarFiltroTratamiento`.

```
137 manejarFiltroTratamiento(tratamiento: TratamientoNodo) {
138   if (this.filtradoTratamiento && this.tratamientoFiltroId == tratamiento.id) {
139     this.filtradoTratamiento = false;
140     this.filtradoTexto = "";
141     this.colorTexto = "#000";
142     this.treeControl.collapseAll();
143     this.limpiarFiltros();
144     this.politicaOriginal();
145   }
146   else {
147     this.aplicarFiltroTratamiento(tratamiento.id)
148     this.filtradoTratamiento = true;
149     this.filtradoAtributo = false;
150     this.tratamientoFiltroId = tratamiento.id;
151     this.filtradoTexto = tratamiento.descripcion;
152     this.colorTexto = tratamiento.color_primario;
153     this.treeControl.dataNodes.forEach(
154       nodo => {
155         if (nodo.id == tratamiento.id) {
156           this.treeControl.expand(nodo)
157         } else {
158           this.treeControl.collapse(nodo)
159         }
160       })
161   }
162 }
```

Código 2.27 Función para manejo de filtro de tratamiento de datos

Por otro lado, el Código 2.29 es donde se realiza el filtrado de anotaciones. En concreto, se verifica que una anotación contiene el tratamiento seleccionado para el filtrado. En caso de éxito se presenta esta anotación, caso contrario se la descarta. Una lógica similar se sigue para el filtrado a nivel de atributo.

El código *frontend* y el manual de usuario de las herramientas de anotación y visualización se encuentra en el ANEXO H y ANEXO I respectivamente. El código completo del *backend* se encuentra en el ANEXO J.

```

41 <mat-tree-node
42   *matTreeNodeDef="let node; when: hasChild"
43   matTreeNodePadding class="fondo"
44   [ngStyle]="{
45     'background-image': estiloTratamiento(node),
46     'border-color': node.color_primario}">
47   <div style="display: flex; justify-content: space-between; width: 95%;">
48     <span
49       matTooltip="Filtrar por este tratamiento"
50       class="texto"
51       (click)="manejarFiltroTratamiento(node)">
52       {{node.descripcion}}
53     </span>
54     <span
55       class="anotaciones"
56       [ngStyle]="{'background-color': node.color_primario}">
57       {{node.numero_anotaciones}}
58     </span>
59   </div>
60 </mat-tree-node>

```

Código 2.28 Código HTML para filtrado de anotaciones por tratamiento de datos

```

69 aplicarFiltroTratamiento(tratamiento_id: number) {
70   let politicaTexto: string = "";
71
72   this.presentacion.politica.parrafos.forEach(
73     parrafo => {
74       let parrafoCss: string = "";
75
76       if (parrafo.titulo != '') {
77         parrafoCss += this.darEstiloTitulo(parrafo)
78       }
79
80       parrafoCss += parrafo.texto_html
81
82       for (let anotacion of parrafo.anotaciones) {
83         for (let valor of anotacion.tratamientos) {
84           if (valor.tratamiento_id === tratamiento_id) {
85             let textoCss: string = this.darEstiloAnotacion(anotacion, valor.color_primario)
86             parrafoCss = parrafoCss.replace(anotacion.texto_html.trim(), textoCss)
87             break;
88           }
89         }
90       }
91       politicaTexto += parrafoCss;
92     }
93   )
94   this.presentarPolitica(politicaTexto);
95 }
96

```

Código 2.29 Función para filtrar anotaciones por tratamiento de datos

3. VERIFICACIÓN Y VALIDACIÓN

El objetivo de este capítulo es presentar la evidencia del funcionamiento de la herramienta de anotación y de visualización. Se divide en dos secciones: 1) en pruebas funcionales de la herramienta donde se valida que los criterios de aceptación de cada historia de usuario se cumplen y 2) el funcionamiento general de la herramienta a través de la anotación de 5 políticas de privacidad de distintas aplicaciones llevadas a cabo por dos anotadores.

Para el desarrollo de este capítulo, fue necesario alojar las herramientas en un servidor. Se hizo uso del servicio en la nube AWS³⁴, los servicios usados son:

1. La base de datos fue alojada en una instancia RDS³⁵
2. El *backend* y *frontend* de la aplicación fueron alojados en una instancia de un servidor EC2³⁶. Se utilizó NGINX³⁷ como servidor web para el *frontend*, mientras que para el *backend* se hizo uso de Gunicorn³⁸ para provisionar la aplicación de Python y de NGINX para realizar un proxy reverso hacia la instancia de Gunicorn que ejecuta la aplicación. Esto se resume en la Figura 3.1.

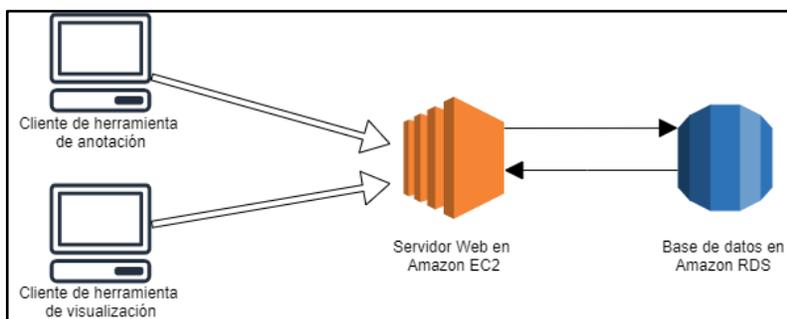


Figura 3.1 Diagrama de despliegue en AWS

La aplicación es servida en las direcciones presentadas en la Tabla 3.1.

³⁴ **AWS:** Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 175 servicios integrales de centros de datos a nivel global.

³⁵ **RDS:** Amazon Relational Database Service (Amazon RDS) es un servicio web que facilita la configuración, el funcionamiento y el escalado de una base de datos relacional en AWS Cloud.

³⁶ **EC2:** Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores.

³⁷ **NGINX:** es un famoso software de servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Sin embargo, hoy en día también sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP.

³⁸ **Gunicorn:** 'Green Unicorn' es un servidor HTTP WSGI de Python para UNIX. El servidor Gunicorn es ampliamente compatible con varios *frameworks* web, implementado de manera simple y ligero en los recursos del servidor.

Tabla 3.1 Direcciones web de la herramienta

Componente	URL
Backend	backend.epn.herramientapoliticaprivacidad.com
Frontend anotación	anotacion.epn.herramientapoliticaprivacidad.com
Frontend visualización	visualizacion.epn.herramientapoliticaprivacidad.com

3.1. PRUEBAS FUNCIONALES

Los resultados presentados en esta sección son las pruebas realizadas por el equipo de desarrollo con el objetivo de comprobar que los criterios de aceptación de cada historia de usuario se cumplen. En la Tabla 3.2 se resume si las historias de usuario fueron terminadas, si cumplen con los criterios de aceptación y si fueron aceptadas por el *product owner*. Esto también puede observarse en el tablero Trello usado para el proyecto³⁹.

Tabla 3.2 Resumen de cumplimiento de las historias de usuario

#	Descripción	Terminada	Criterios de aceptación	¿Aceptada por el product owner?
HU-01	Inicio y cierre de sesión	Si	Alcanzados	Si
HU-02	Administración de usuarios en la herramienta	Si	Alcanzados	Si
HU-03	Administración de tratamientos de datos	Si	Alcanzados	Si
HU-04	Administración de atributos	Si	Alcanzados	Si
HU-05	Administración de valores	Si	Alcanzados	Si
HU-06	Administración de políticas de privacidad	Si	Alcanzados	Si
HU-07	Visualización de políticas de privacidad por anotar	Si	Alcanzados	Si
HU-08	Anotación de una política de privacidad	Si	Alcanzados	Si
HU-09	Visualización, edición y eliminación de anotaciones	Si	Alcanzados	Si
HU-10	Notificación de inconsistencia en anotación	Si	Alcanzados	Si
HU-11	Visualización de políticas de privacidad para consolidación	Si	Alcanzados	Si
HU-12	Cálculo de coeficiente inter-anotador	Si	Alcanzados	Si
HU-13	Consolidación de una política de privacidad	Si	Alcanzados	Si
HU-14	Visualización, edición y eliminación de anotaciones en consolidación	Si	Alcanzados	Si
HU-15	Visualización y copia de anotaciones de la etapa de anotación en consolidación	Si	Alcanzados	Si

³⁹ <https://trello.com/b/N8Ntg6om/anotaci%C3%B3n-de-pol%C3%A9ticas-de-privacidad>

HU-16	Notificación de sección inconsistente en anotación	Si	Alcanzados	Si
HU-17	Página de bienvenida en herramienta de visualización	Si	Alcanzados	Si
HU-18	Lista de políticas de privacidad para visualizar	Si	Alcanzados	Si
HU-19	Visualización de una política de privacidad anotada	Si	Alcanzados	Si
HU-20	Filtrado de anotaciones en visualización	Si	Alcanzados	Si

Ahora se muestra la evidencia de cada historia de usuario y sus criterios de aceptación.

3.1.1. MÓDULO DE ADMINISTRACIÓN Y AUTENTICACIÓN - *SPRINT* 1

Esta sección corresponde a las pruebas de las funcionalidades implementadas en el *sprint* 1, las cuales son: inicio y cierre de sesión, y administración de los elementos: usuarios, políticas, tratamientos, atributos y valores.

3.1.1.1. HU-01: Inicio y cierre de sesión

Para la historia de usuario HU-01 los criterios de aceptación son:

1. Como anotador o administrador requiero estar en la capacidad de iniciar sesión al ingresar mi usuario (correo electrónico) y contraseña.
2. Como anotador o administrador requiero estar en la capacidad de cerrar sesión.

Para validar este funcionamiento se ingresó a la herramienta de anotación y en la pantalla de ingreso presentada en la Figura 3.2, se llenó el formulario con el usuario y contraseña previamente almacenados en la base de datos. Al hacer *click* en ingresar se obtiene la pantalla de bienvenida de la herramienta presentada en la Figura 3.3.

The image shows a login form titled "Herramienta Anotacion". It contains two input fields. The first is labeled "Email de usuario *" and contains the text "felix.minom@gmail.com". The second is labeled "Contraseña" and contains a series of dots representing a masked password. Below the input fields is a red button with the text "Ingresar".

Figura 3.2 Formulario de inicio de sesión

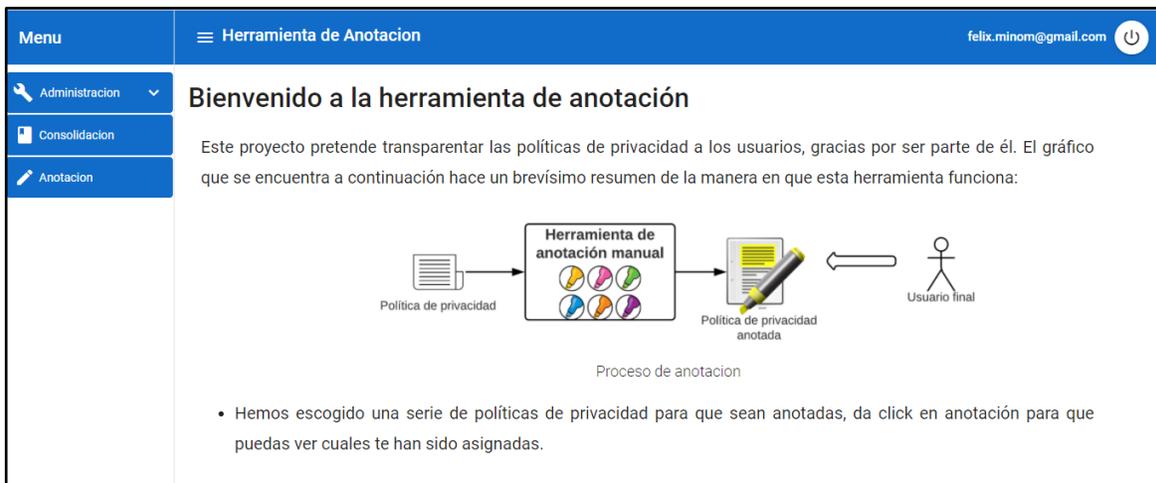


Figura 3.3 Pantalla de bienvenida herramienta anotación

El usuario que inició una sesión tiene un rol de administrador. Como se observa en el menú lateral de la Figura 3.3 este usuario tiene acceso a los módulos de: administración, consolidación y anotación. También en la esquina superior derecha se encuentra el nombre del usuario que ingresó a la herramienta y un botón para cerrar la sesión. Al hacer *click* en este botón se obtiene la pantalla de inicio que se presentó en la Figura 3.2. Por lo tanto, se cumplen los criterios de aceptación de HU-01.

3.1.1.2. HU-02: Administración de usuarios en la herramienta

Como administrador de la herramienta de anotación requiero:

1. Administrar los roles de usuario y módulos referentes a la anotación.

Para probar la administración de estos elementos se usó la herramienta Postman⁴⁰. La estructura de una petición en Postman se presenta en la Figura 3.4:

1. Método HTTP: describe el método a ejecutar, es decir: GET, POST, PATCH o DELETE.
2. URL: dirección del servicio web que se desea utilizar.
3. Cuerpo de la petición: son los parámetros en formato JSON necesarios para la ejecución del método HTTP. Este campo es opcional.
4. Respuesta de la petición: son datos en formato JSON que generalmente se ajustan a modelos de datos.

⁴⁰ **Postman**: es una plataforma de colaboración para el desarrollo de API. Es posible enviar de forma rápida y sencilla solicitudes de REST, SOAP y GraphQL directamente desde *Postman*.

5. Estado de la respuesta: representa un código HTTP de respuesta.



Figura 3.4 Estructura de una petición y creación de un módulo a través de *Postman*

A continuación, se presentan los resultados obtenidos al realizar las acciones de creación, consulta, edición y eliminación de un módulo. Los resultados de estas acciones para el rol de usuario son presentados en el ANEXO K.

La Figura 3.4 corresponde a la creación de un módulo. En este caso se obtuvo un código de respuesta 201 y una respuesta con mensaje de éxito de la operación.

En la Figura 3.5 se presenta la consulta del módulo previamente creado. El `id` del módulo es enviado como parte de la URL.

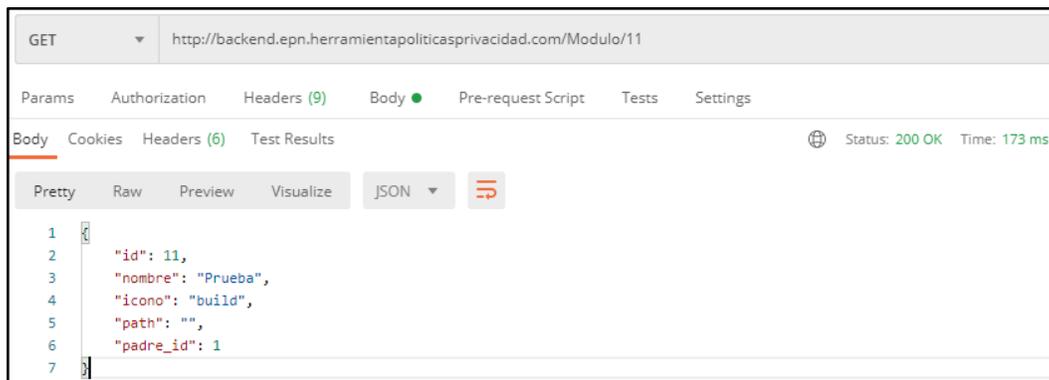


Figura 3.5 Consulta de módulo a través de *Postman*

En la Figura 3.6 se presenta el resultado de realizar una acción PATCH sobre el módulo creado para esta prueba. El atributo modificado fue `path`, el cual previamente era un campo nulo. La siguiente ocasión que se consulte este módulo, el atributo `path` tendrá el valor `"/path/prueba"`.

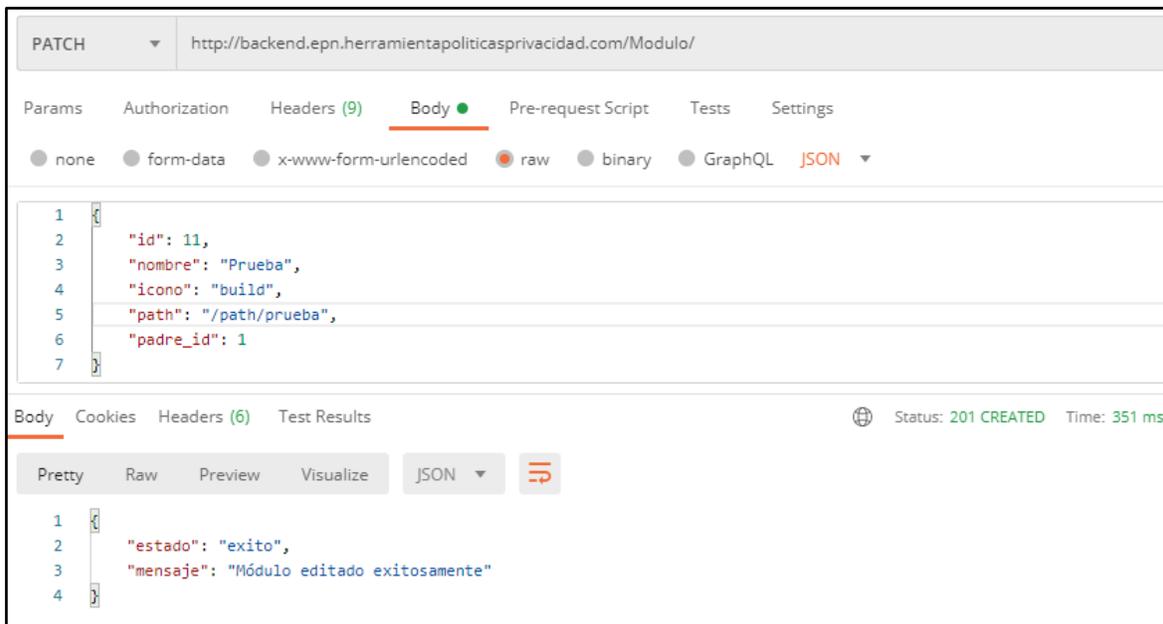


Figura 3.6 Modificación de módulo a través de *Postman*

Finalmente, la Figura 3.7 se realiza la eliminación del módulo creado con fines de prueba. El `id` del módulo es enviado a través de la URL.

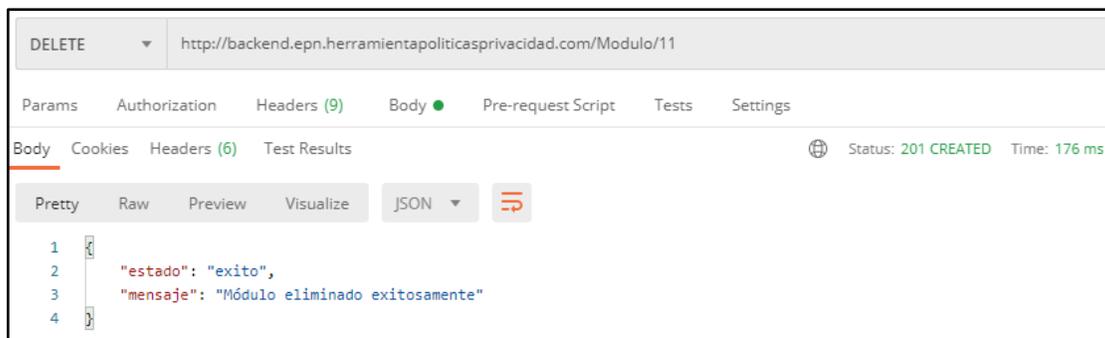


Figura 3.7 Eliminación de módulo a través de *Postman*

2. Crear un usuario. El usuario debe tener los campos: correo electrónico, rol de usuario y entrenamiento. No debe permitirse campos en blanco.
3. Elegir el rol de usuario (anotador o administrador) al momento de crear uno.
4. Elegir si un usuario estará en entrenamiento o no.
5. Al momento de crear un usuario, este nuevo usuario debe ser notificado a su correo electrónico acerca de su rol de usuario en la herramienta y sus credenciales de acceso. La clave será autogenerada.

Los criterios de aceptación 2, 3 y 4 se pueden validar al momento de crear un usuario. En la Figura 3.8 se presenta el formulario que se despliega para crear un usuario, así mismo se presenta el mensaje de error del formulario si contiene algún campo en blanco.

The figure displays two versions of a 'Creación de usuario' (User Creation) form. The left version, labeled 'Formulario con campos vacíos', shows a red error message at the top: 'El formulario contiene campos vacíos'. The form fields are empty, and the 'Email de Usuario' field is highlighted with a red border. The right version, labeled 'Formulario sin errores', shows the form with the email 'felix.mino@epn.edu.ec' entered, the 'Anotador' role selected, and the 'En entrenamiento' toggle switch turned on. Both forms have a 'Guardar' button at the bottom right.

Figura 3.8 Formulario creación de usuario

Por otro lado, el criterio de aceptación 5 es comprobado mediante la Figura 3.9, donde se presenta el correo recibido por el usuario registrado en la herramienta.

The image shows a welcome email template. At the top, it reads 'SISTEMA DE ANOTACION DE POLITICAS DE PRIVACIDAD EN ESPAÑOL'. Below that is a blue header with 'BIENVENIDO!'. The main body of the email says: 'Has sido registrado como Anotador en el sistema de anotación de políticas de privacidad en español. A continuación podrás encontrar tus credenciales de acceso.' It then lists the user as 'felix.mino@epn.edu.ec' and the password as 'WTMx4VHJ'. There is a blue 'Ingresar' button. At the bottom, it says 'Si esto no funciona, copia y pega la siguiente url en tu buscador:' followed by the URL 'http://anotacion.epn.herramientapoliticaprivacidad.com/'. The final line says 'Si tienes alguna duda o sugerencia no dudes en responder a este'.

Figura 3.9 Correo recibido al ser registrado como anotador de la herramienta.

6. Visualizar una lista de los usuarios presentes en la herramienta.

Luego que el usuario fue creado en la prueba de funcionamiento previa, se puede observar la lista de usuarios que son parte de la herramienta. Esto se presenta en la Figura 3.10.

Gestión de Usuarios						
Lista, creación, eliminación y edición de usuarios						+Nuevo
Filtrar usuario por email						
#	Usuario	Rol de usuario	Activo	En entrenamiento	Editar	Eliminar
1	felix.minom@gmail.com	Administrador	✓	✗		
2	felix.mino@epn.edu.ec	Anotador	✓	✓		

Figura 3.10 Lista de usuarios en la herramienta

7. Editar los atributos de un usuario: correo electrónico, rol de usuario, si el usuario se encuentra en entrenamiento o no y si el usuario está activo o no en la herramienta.

A través de la Figura 3.11 se evidencia la edición del campo “en entrenamiento” del usuario creado previamente. Luego en la Figura 3.12 se presenta la lista actualizada de usuarios después de esta edición, como se puede observar el usuario 2 ahora ya no se encuentra en entrenamiento.

Edición de usuario	Edición de usuario
<p>Email de Usuario felix.mino@epn.edu.ec</p> <p>Escoja el rol de usuario:</p> <p><input type="radio"/> Administrador <input checked="" type="radio"/> Anotador</p> <p>En entrenamiento: <input checked="" type="checkbox"/></p> <p>Activo: <input checked="" type="checkbox"/></p> <p>Guardar </p>	<p>Email de Usuario felix.mino@epn.edu.ec</p> <p>Escoja el rol de usuario:</p> <p><input type="radio"/> Administrador <input checked="" type="radio"/> Anotador</p> <p>En entrenamiento: <input type="checkbox"/></p> <p>Activo: <input checked="" type="checkbox"/></p> <p>Guardar </p>
Formulario antes de edición	Formulario después de edición

Figura 3.11 Formulario de edición de usuario

Gestión de Usuarios						
Lista, creación, eliminación y edición de usuarios						+Nuevo
Filtrar usuario por email _____						
#	Usuario	Rol de usuario	Activo	En entrenamiento	Editar	Eliminar
1	felix.minom@gmail.com	Administrador	✔	✘		
2	felix.mino@epn.edu.ec	Anotador	✔	✘		

Figura 3.12 Lista de usuarios luego de edición

8. Eliminar un usuario. En caso de que no sea posible eliminarlo, se requiere marcarlo como usuario inactivo, lo que restringirá el acceso a la herramienta.

Antes de proceder a eliminar el usuario creado con fines de prueba, se editó el campo “activo” a falso, de esta manera el usuario perdió el acceso a la herramienta. Esto se evidencia en la Figura 3.13.

Herramienta Anotacion

Email de usuario *

Contraseña

Error en el usuario o contraseña!

Ingresar

Figura 3.13 Intento de acceso con usuario inactivo.

Para verificar la integridad de clave foránea se realizó una anotación de una política creada al azar. De esta manera, al momento de eliminar el usuario se obtiene el mensaje presentado en la Figura 3.14.

Gestión de Usuarios						
Lista, creación, eliminación y edición de usuarios						+Nuevo
Filtrar usuario por email _____						
#	Usuario	Rol de usuario	Activo	En entrenamiento	Editar	Eliminar
1	felix.minom@gmail.com	Administrador	✔	✔		

Figura 3.14 Prueba de eliminación de usuario manteniendo integridad de clave foránea

Una vez comprobado esto, se procedió a eliminar al usuario 2. Al presionar el botón eliminar de la Figura 3.12, la herramienta está en la capacidad de eliminar a este usuario ya que no realizó ninguna anotación. En la Figura 3.15 se presenta el mensaje luego de la eliminación y la lista de usuarios actualizada.

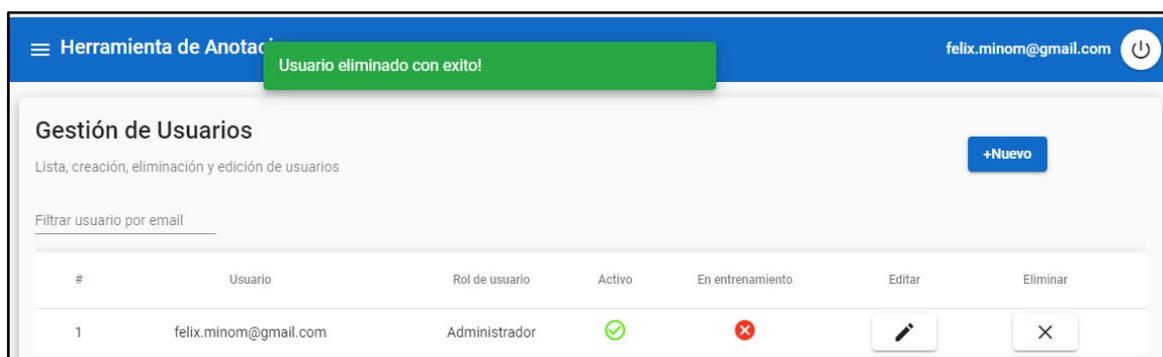


Figura 3.15 Prueba de eliminación de usuario

3.1.1.3. HU-03: Administración de tratamientos de datos

Como administrador de la herramienta de anotación requiero:

1. Administrar los colores haciendo llamada a los respectivos APIs RESTFul.

Para realizar estas pruebas se hizo uso de la herramienta Postman. Los resultados se presentan en el ANEXO K.

2. Crear un tratamiento de datos. El tratamiento de datos debe tener: descripción y color. No deben permitirse campos en blanco.
3. Visualizar los tratamientos de datos existentes.

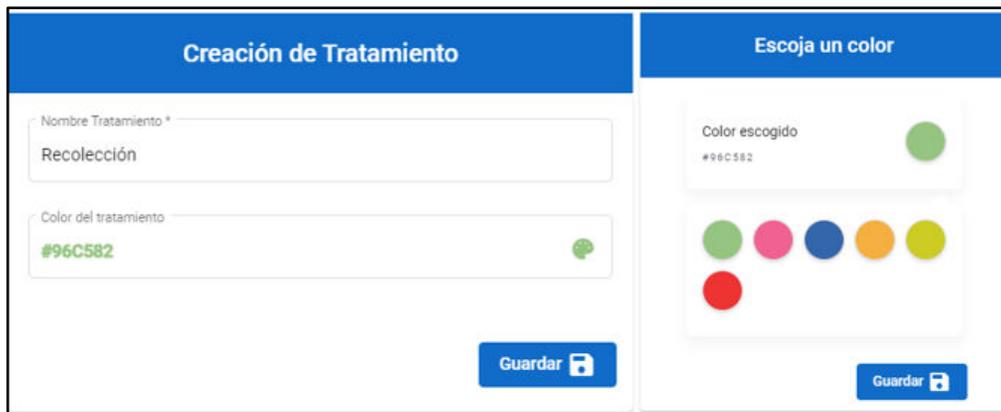
Los criterios de aceptación 2 y 3, son validados a través de la creación de un tratamiento. En la Figura 3.16 se presentan los diálogos para la creación de un tratamiento y para la asignación de un color a un tratamiento. En la Figura 3.17 se presenta la lista de tratamientos luego de guardar el tratamiento presentado en la figura anterior.

4. Editar un tratamiento de datos: descripción y color.

En la Figura 3.18 se presenta el diálogo para edición de tratamiento. El resultado se evidencia en la Figura 3.19. Se modificó la descripción y el color del tratamiento previamente creado.

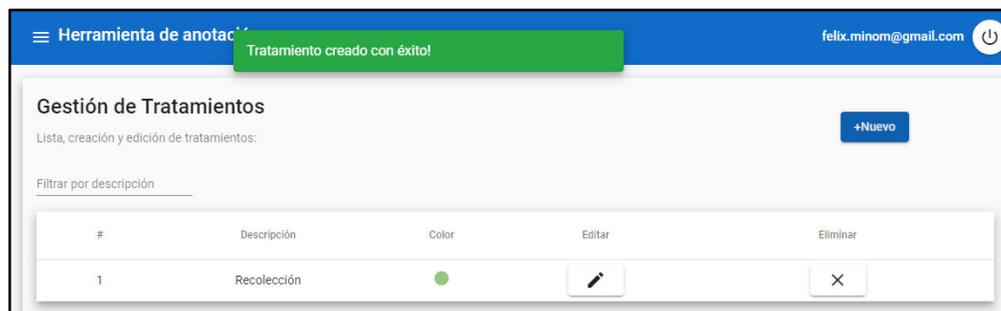
5. Eliminar un tratamiento de datos. Manteniendo la integridad referencial.

Para probar que la integridad referencial se está respetando, se creó un atributo de prueba bajo el tratamiento previamente creado, esto debería prevenir la eliminación del tratamiento ya que su clave primaria es usada como clave foránea en otro registro (atributo). La creación del atributo se evidencia en la Figura 3.20. El intento de eliminación genera un error, esto se presenta en la Figura 3.21.



El formulario está dividido en dos secciones. La sección 'Creación de Tratamiento' contiene un campo de texto para 'Nombre Tratamiento *' con el valor 'Recolección' y un campo de color para 'Color del tratamiento' con el código hexadecimal '#96C582'. La sección 'Escoja un color' muestra 'Color escogido #96C582' con un círculo verde y una paleta de colores con opciones roja, verde, rosa, azul, naranja y amarillo. Ambas secciones tienen un botón 'Guardar' con un ícono de disco.

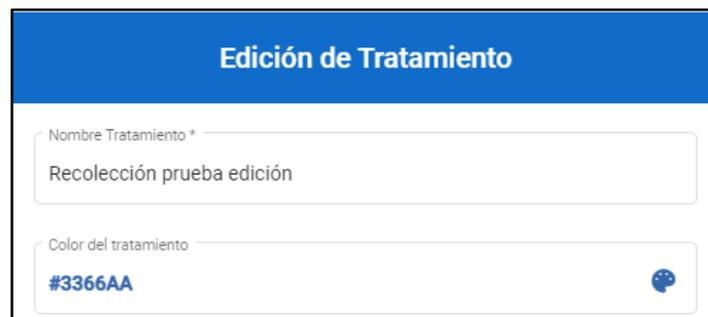
Figura 3.16 Formulario para creación de tratamiento y asignación de color



La interfaz muestra una barra superior con 'Herramienta de anotación', un mensaje de éxito 'Tratamiento creado con éxito!' y el correo 'felix.minom@gmail.com'. El título principal es 'Gestión de Tratamientos' con un botón '+Nuevo'. Debajo hay un filtro 'Filtrar por descripción' y una tabla con una sola fila de datos.

#	Descripción	Color	Editar	Eliminar
1	Recolección	●		

Figura 3.17 Lista de tratamientos



El formulario 'Edición de Tratamiento' tiene un campo de texto para 'Nombre Tratamiento *' con el valor 'Recolección prueba edición' y un campo de color para 'Color del tratamiento' con el código hexadecimal '#3366AA' y un ícono de paleta de colores.

Figura 3.18 Prueba de edición de tratamiento

#	Descripción	Color
1	Recolección prueba edición	●

Figura 3.19 Lista de tratamientos luego de edición

Herramienta de anotación felix.minom@gmail.com

Atributo creado con éxito!

Gestión de atributos

Lista, creación, eliminación y edición de atributos:

Escoja un tratamiento de datos *

Recolección prueba edición

[+Nuevo](#)

#	Descripción	Color	Editar	Eliminar
1	Tipo de dato	●		

Figura 3.20 Creación de atributo para prueba de integridad referencial

Herramienta de anotación felix.minom@gmail.com

ERROR eliminando tratamiento!

Gestión de Tratamientos

Lista, creación y edición de tratamientos:

Filtrar por descripción

#	Descripción	Color	Editar	Eliminar
1	Recolección prueba edición	●		

Figura 3.21 Prueba de eliminación de tratamiento que contiene atributos

Por otro lado, cuando el tratamiento no tiene conflictos de integridad referencial es eliminado exitosamente, esto se evidencia en la Figura 3.22.

Herramienta de anotación felix.minom@gmail.com

Tratamiento eliminado con éxito!

Gestión de Tratamientos

Lista, creación y edición de tratamientos:

Filtrar por descripción

#	Descripción	Color	Editar	Eliminar
---	-------------	-------	--------	----------

Figura 3.22 Prueba de eliminación tratamiento que no contiene atributos

6. Filtrar un tratamiento de datos en base a la descripción.

Para probar el funcionamiento del filtro de tratamientos, se crearon los tratamientos de datos de la Tabla 1.3, esto se presenta en la Figura 3.23. Luego en el campo “Filtrar por descripción” se colocó el valor “Recolección” por lo que este debería ser el único tratamiento en la lista, esto se evidencia en la Figura 3.24.

#	Descripción	Color	Editar	Eliminar
1	Recolección	●		
2	Conservación	●		
3	Transferencia	●		
4	Control del titular	●		

Figura 3.23 Lista de tratamientos para prueba de filtrado

Filtrar por descripción				
Recolección				
#	Descripción	Color	Editar	Eliminar
1	Recolección	●		

Figura 3.24 Lista de tratamientos filtrada

3.1.1.4. HU-04 hasta HU-06

Los criterios de aceptación de estas historias de usuario corresponden a la administración de atributos, valores y políticas de privacidad respectivamente. Las pruebas de funcionalidad con sus evidencias se presentan en el ANEXO K.

3.1.2. MÓDULO DE ANOTACIÓN, CONSOLIDACIÓN Y VISUALIZACIÓN - SPRINT 2, 3 y 4

En esta sección se validaron los criterios de aceptación desde la historia de usuario HU-07 hasta HU-20. Mediante un caso de prueba bajo condiciones controladas se verificó que todos los criterios de aceptación están garantizados.

El caso de prueba consiste en la anotación de dos políticas de privacidad generadas con texto *lorem ipsum* ⁴¹. Cada política de privacidad está compuesta de tres secciones. Fueron anotadas y consolidadas por el desarrollador de la herramienta. En la Figura 3.25 se observa la estructura del caso de prueba.

⁴¹ **Lorem ipsum:** es el texto que se usa habitualmente en diseño gráfico en demostraciones de tipografías o de borradores de diseño para probar el diseño visual antes de insertar el texto final.

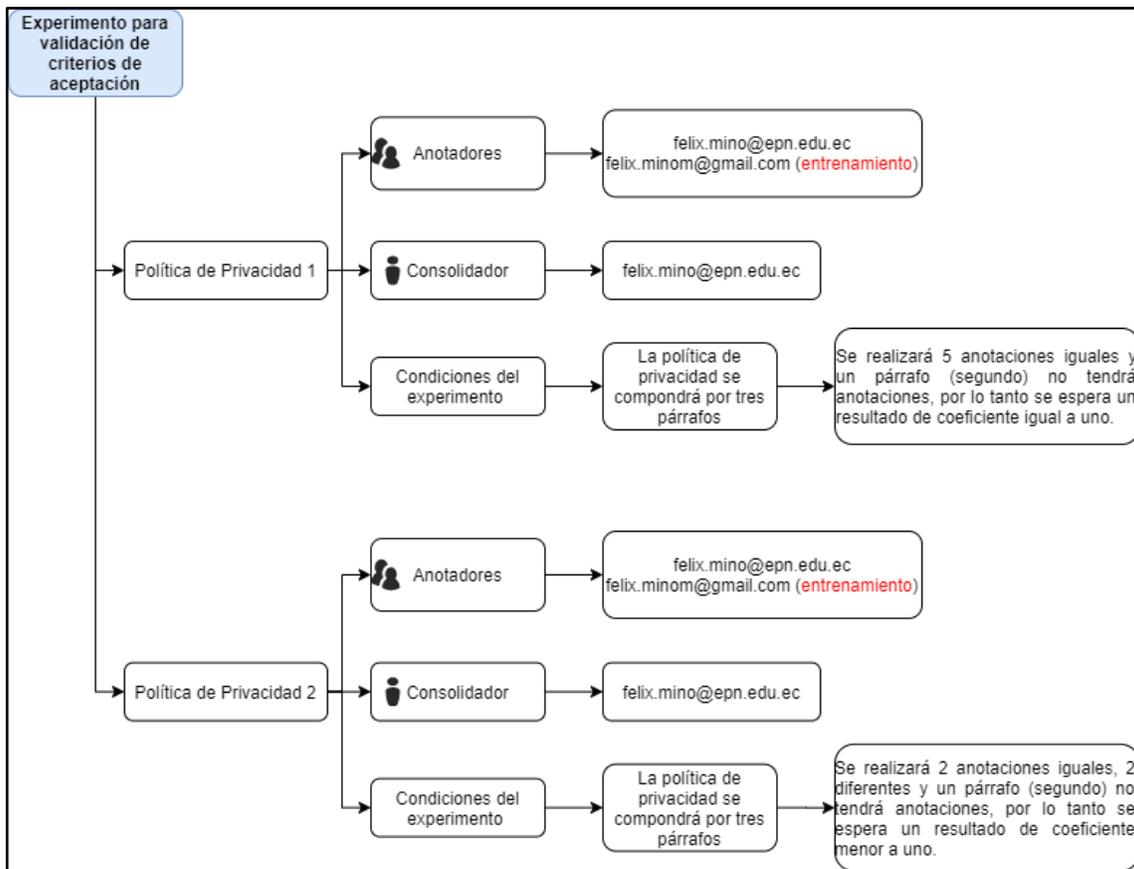


Figura 3.25 Estructura del caso de prueba para pruebas funcionales de anotación

Por otro lado, los tratamientos disponibles para realizar la anotación de las dos políticas de privacidad son los presentados en la Tabla 1.3. Se realizó una elección aleatoria de estos tratamientos para realizar las anotaciones de las políticas de privacidad. En la Tabla 3.3 se resume las anotaciones realizadas

Tabla 3.3 Anotaciones realizadas sobre políticas de privacidad en caso de prueba

	Usuarios	Sec ción	Anotación	Tratamiento	Atributo	Valor	Acción	
Política de privacidad 1	1* 2**	1	1.1	Recolección	Propósito	Genérico	Ejecuta	
				Conservación	Propósito	Genérico		
			1.2	Transferencia	Transferencia internacional	Genérico	No ejecuta	
			1.3	Conservación	Período de conversación	Genérico	Ejecuta	
				Control del titular	Eliminación	Genérico		
		2	Este párrafo no contendrá anotaciones					
		3	1.4	Recolección	Tipo de dato	Genérico	Ejecuta	
				Conservación	Propósito	Genérico		
				Control del titular	Acceso	Genérico		
		1.5	Transferencia	Propósito	Genérico	Ejecuta		

Política de privacidad 2	1*	1	2.1	Recolección	Origen de los datos	Genérico	No ejecuta
				Transferencia	Transferencia internacional	Genérico	
			2.2	Control del titular	Edición	Genérico	Ejecuta
		2	Este párrafo no contendrá anotaciones				
		3	2.3	Transferencia	Propósito	Genérico	Ejecuta
			2.4	Conservación	Propósito	Genérico	Ejecuta
			Control del titular	Edición	Genérico		
	2**	1	2.5	Recolección	Elección de usuario	Genérico	No ejecuta
				Transferencia	Transferencia internacional	Genérico	
			2.6	Control del titular	Edición	Genérico	Ejecuta
		2	Este párrafo no contendrá anotaciones				
		3	2.7	Transferencia	Propósito	Genérico	Ejecuta
			2.8	Conservación	Propósito	Genérico	No ejecuta
				Control del titular	Edición	Genérico	

* El usuario "felix.mino@epn.edu.ec" realizó las anotaciones. Así también, estas anotaciones sirven para consolidar la política de privacidad.

** El usuario "felix.minom@gmail.com" realizó las anotaciones.

Es importante mencionar que en la "política de privacidad 2" las anotaciones diferentes son 2.1 respecto a 2.5 y 2.4 respecto a 2.8.

A continuación, se presentan los criterios de aceptación y las evidencias de su funcionamiento de las diferentes historias de usuario.

3.1.2.1.1. HU-07: Visualización de políticas de privacidad por anotar

Como anotador o administrador de la herramienta requiero:

1. Visualizar las políticas de privacidad que me fueron asignadas para anotar y aún no fueron terminadas.
2. Junto a cada política de privacidad visualizar el porcentaje de progreso.

Al dar *click* sobre la pestaña "Anotación" del menú lateral se despliega la lista de políticas que un usuario tiene pendiente por anotar. La lista de políticas por anotar presentadas en la Figura 3.26 corresponden al usuario "felix.minom@gmail.com" y el progreso de éstas es 100% ya fueron totalmente anotadas.

Herramienta de anotación		felix.minom@gmail.com 
Políticas de privacidad por anotar		
Lista de políticas de privacidad asignadas para codificar:		
#	Políticas asignadas para codificar	Progreso
1	Política de privacidad 1	100%
2	Política de privacidad 2	100%

Figura 3.26 Lista de políticas de privacidad por anotar

3.1.2.1.2. HU-08: Anotación de una política de privacidad

Como anotador o administrador de la herramienta requiero:

1. Al dar *click* en una política de privacidad en la sección visualización ser redirigido a la página para la anotación de la misma.
2. Tener una lista de los tratamientos de datos, sus atributos y sus valores. Solo se presentarán los tratamientos que incluyan al menos un atributo y un valor.

Estos dos criterios de aceptación pueden comprobarse al dar *click* en una de las políticas de la lista presentada en la Figura 3.26, en este caso “Política de privacidad 1”. Como se observa en la Figura 3.27, se presenta la pantalla de anotación para la política antes mencionada, así como también la lista de tratamientos de datos de la Tabla 1.3.

Para comprobar que se incluyan sólo tratamientos de datos que contengan al menos un atributo y un valor se realizó lo siguiente: se creó un tratamiento de datos con la descripción “Prueba”, se cargó nuevamente la página de anotación y se comprobó que el tratamiento no existía en la lista. Luego, se creó un atributo con la misma descripción del tratamiento y se comprobó que no exista en la lista. Finalmente, se creó un valor para el atributo “Prueba” con la misma descripción y se comprobó que exista en la lista de tratamientos disponibles. Esto se evidencia desde la Figura 3.28 hasta la Figura 3.33.

3. Observar la política separada por las secciones que fueron predefinidas.

Esto se puede comprobar en la Figura 3.27. En la sección de anotación existen flechas para avanzar y retroceder y en este caso la política de privacidad tiene tres secciones.

4. Guardar una anotación. Una anotación consiste en la combinación de: uno o más valores, una sección de texto, un valor que identifica si la anotación ejecuta o no las

acciones presentadas en la política y un comentario (opcional), sin embargo, el comentario no será cubierto por esta historia de usuario.

En la Figura 3.34 se presenta la interfaz gráfica con el texto, la acción y los valores seleccionados para realizar la anotación 1.1 de la Tabla 3.3.

5. Cuando se haya llegado a la última sección de la política y se de *click* en “siguiente” la herramienta debe notificar que se llegó al final del documento y requiero poder elegir si terminar o no la codificación.

En la Figura 3.36 se observa el cuadro de diálogo que contiene la notificación de finalización de la política de privacidad. En caso de ser aceptada esta política de privacidad no podrá ser modificada en un futuro.



Figura 3.27 Pantalla de anotación



Figura 3.28 Creación de tratamiento de prueba



Figura 3.29 Lista de tratamientos disponibles después de creación de tratamiento

#	Descripción	Color
1	Prueba	●

Figura 3.30 Creación de atributo de prueba



Figura 3.31 Lista de tratamientos disponibles después de creación de atributo

#	Descripción	Color
1	Prueba	●

Figura 3.32 Creación de valor de prueba

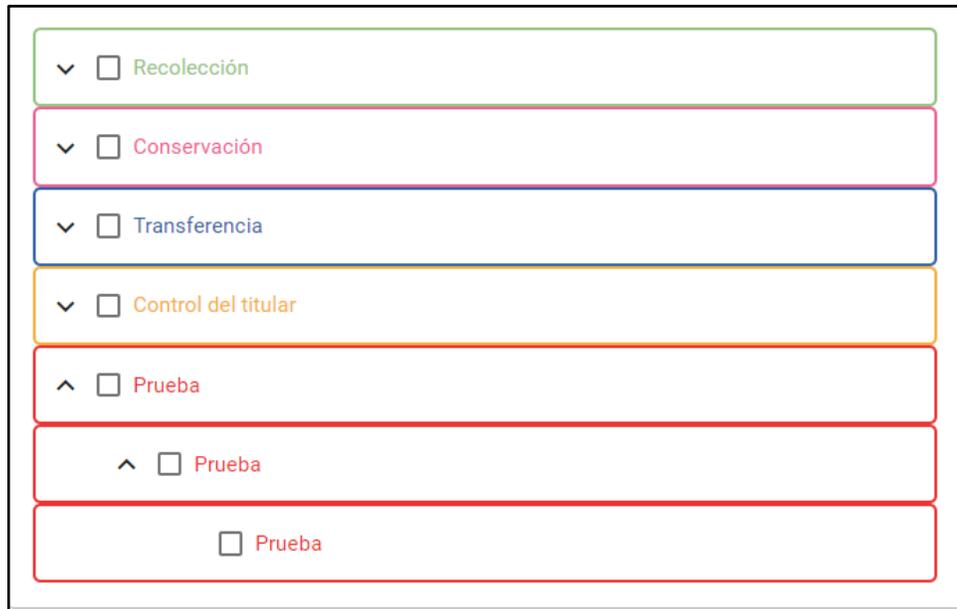


Figura 3.33 Lista de tratamientos disponibles después de creación de valor de prueba

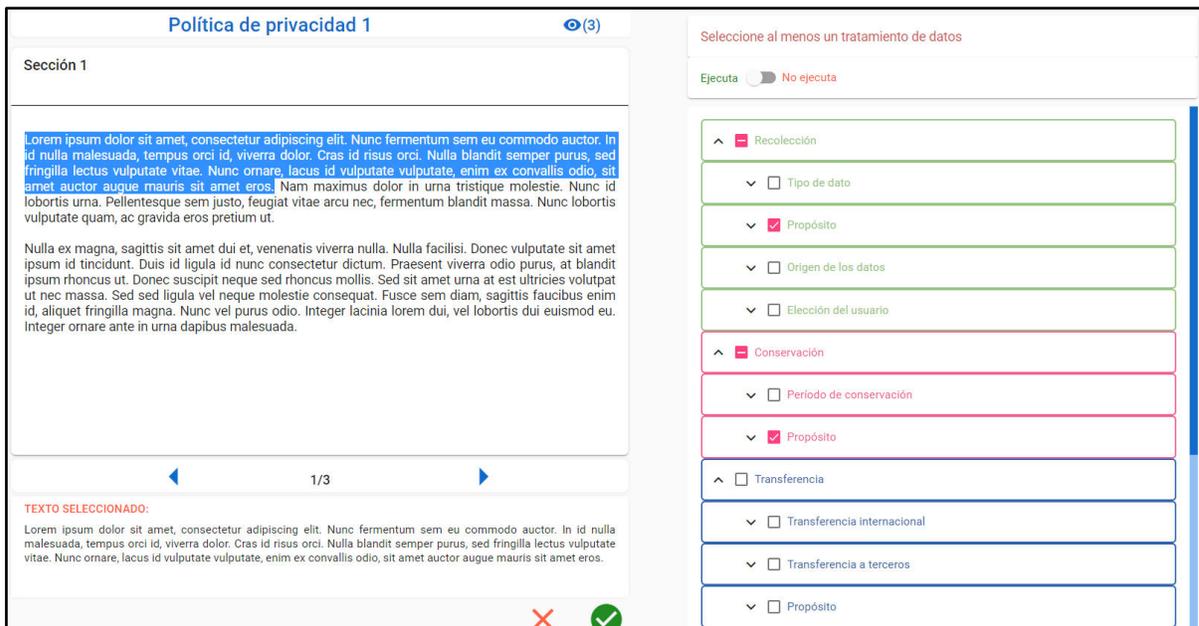


Figura 3.34 Creación de Anotación 1.1

En la Figura 3.35 se observa los registros de la anotación creada desde la interfaz gráfica.

id	texto	consolidar	ejecuta	tratamiento_descripcion	atributo_descripcion	valor_descripcion
2	Lorem ipsum dolor sit amet, consectetur adipisci...	0	1	Recolección	Propósito	Genérico
2	Lorem ipsum dolor sit amet, consectetur adipisci...	0	1	Conservación	Propósito	Genérico

Figura 3.35 Registro de anotación 1.1 en la base de datos



Figura 3.36 Notificación de finalización de política de privacidad

3.1.2.1.3. HU-09: Visualización, edición y eliminación de anotaciones

Como anotador o administrador de la herramienta requiero:

1. Tener un conteo del número de anotaciones totales que realicé sobre una sección de una política de privacidad.

Luego de realizar todas las anotaciones en la sección 1 de la “Política de privacidad 1” se presenta la Figura 3.37, en donde se encuentra el número de anotaciones de la sección al lado derecho del nombre de la política de privacidad.

2. Visualizar las anotaciones que realicé sobre una sección.

Al dar *click* en el conteo de anotaciones se despliega el diálogo donde se muestra el resumen de las anotaciones realizadas sobre la sección 1 de la “Política de privacidad 1”. Esto se presenta en la Figura 3.39.

3. Editar una anotación previamente realizada. Se podrá editar: la acción (ejecuta o no ejecuta), el texto de la anotación y el comentario.

Al dar *click* en el ícono de lápiz de la Figura 3.39, se despliega el diálogo para la edición de una anotación. Como se observa en la Figura 3.38 es posible modificar los valores descritos en el criterio de aceptación.

4. Eliminar una anotación que realicé.

Se creo una anotación con tratamientos de datos aleatorios, como se observa en la Figura 3.40. Al dar *click* en el ícono en rojo de la anotación se despliega un cuadro de confirmación de la eliminación. Esto se presenta en la Figura 3.40.



Figura 3.37 Conteo de anotaciones sobre una sección

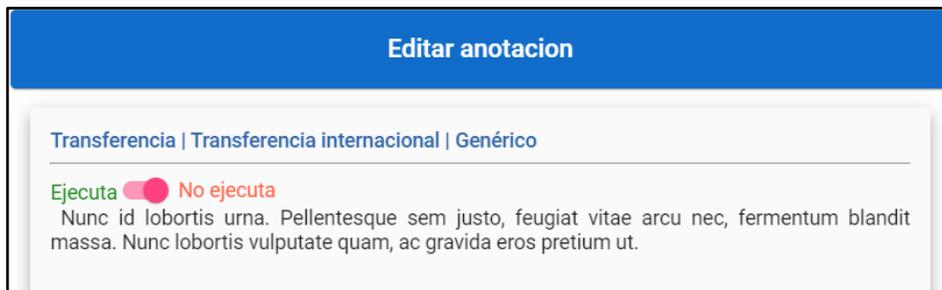


Figura 3.38 Diálogo de edición de anotación

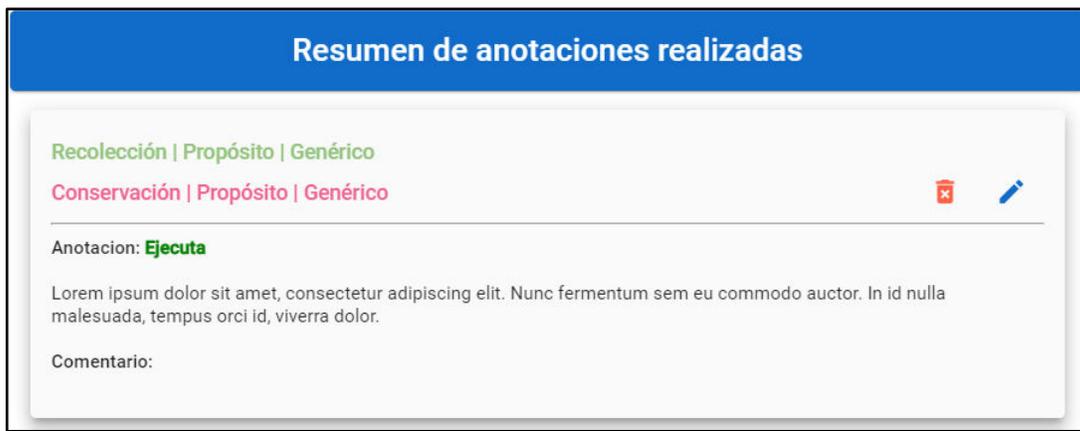


Figura 3.39 Resumen de anotaciones realizadas sobre una sección

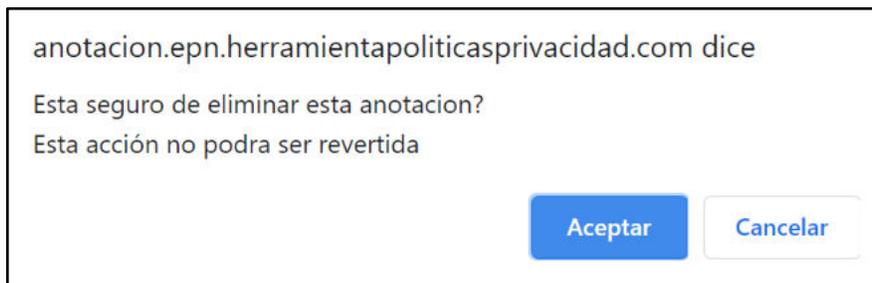


Figura 3.40 Diálogo eliminación de anotación

3.1.2.1.4. HU-10: Notificación de inconsistencia en anotación

Como anotador o administrador (en entrenamiento) de la herramienta requiero:

1. Ser notificado en caso de que uno o más de los valores que estoy usando para realizar la anotación no se encuentren anotados por otro usuario en esa sección de la política. Esto para cada acción (ejecuta o no ejecuta).
2. Tener una lista de posibles tratamientos (sugerencias) para realizar la anotación. Esto se hará en base a valores previamente anotados por otros usuarios. En caso de esa sección haya sido dejada sin anotaciones de manera intencional, en lugar de una lista

de sugerencia de valores, requiero tener una sugerencia de dejar esta sección sin anotar.

Se realizó una anotación con un tratamiento de datos que no exista en la anotación 1.1 de la Tabla 3.3 con el usuario “felix.minom@gmail.com”. Al guardar, se despliega un diálogo de notificación de inconsistencia como se observa en la Figura 3.41.

Para validar el segundo criterio de aceptación se realizó una anotación sobre la segunda sección de “Política de privacidad 1”. Por lo tanto, como se muestra en Figura 3.42 no existen tratamientos de datos sugeridos.

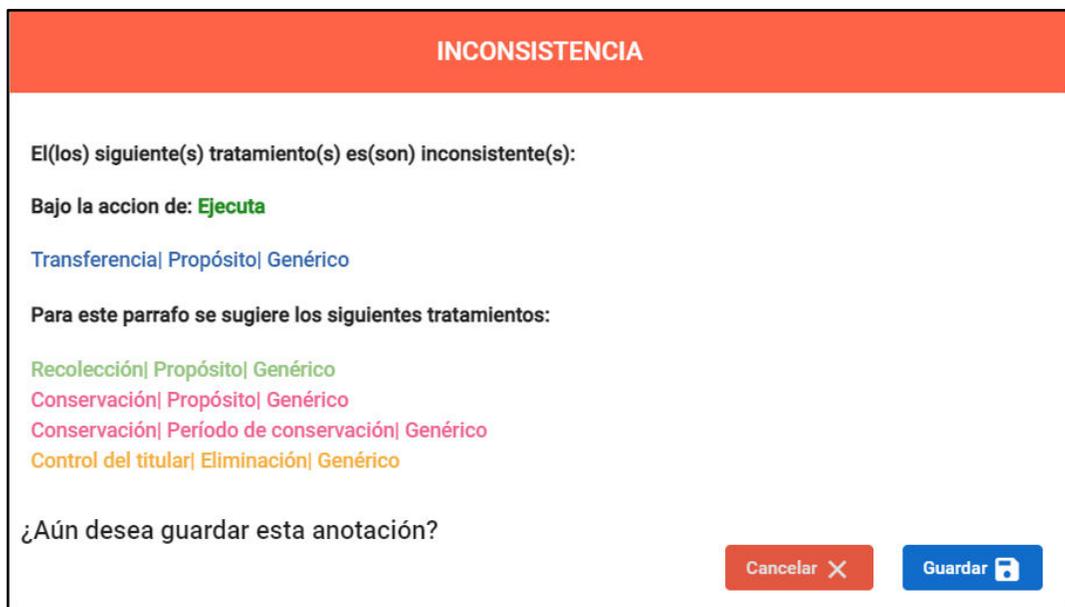


Figura 3.41 Notificación de inconsistencia con tratamientos sugeridos

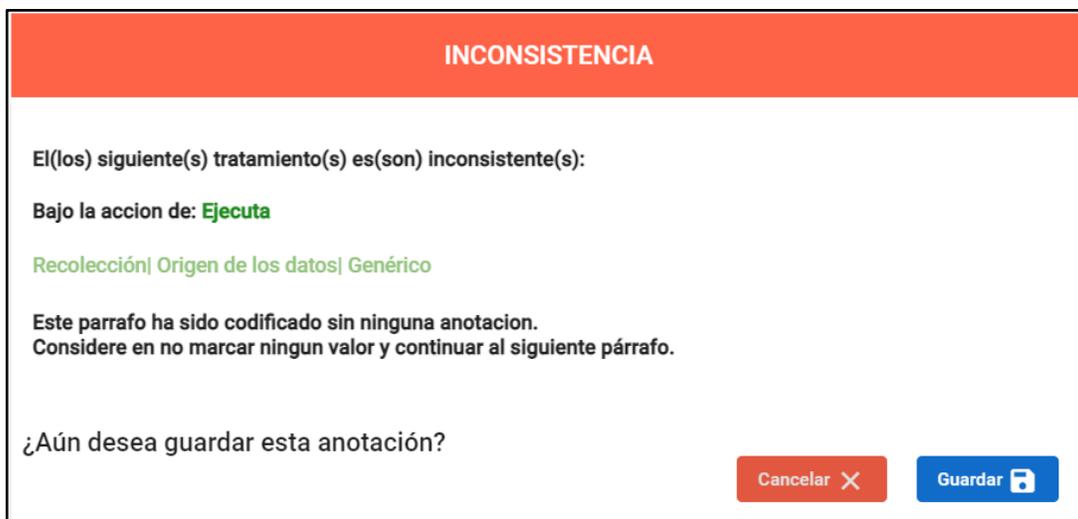


Figura 3.42 Notificación de inconsistencia sin tratamientos sugeridos

3.1.2.1.5. HU-11: Visualización de políticas de privacidad para consolidación

Como administrador de la herramienta requiero:

1. Visualizar en una lista las políticas de privacidad que me fueron asignadas para consolidar y aún no terminadas. Solamente cuando la política fue terminada de anotar por todos los usuarios asignados para esa tarea.
2. Junto a cada política de privacidad visualizar el porcentaje de progreso.

Como se presenta en la Figura 3.43 la lista de políticas listas para consolidar es desplegada. Esto se respalda con la Figura 3.44 donde se presenta que los anotadores finalizaron dicha etapa.

#	Políticas asignadas para codificar	Progreso	Detalles
1	Política de privacidad 1	100%	☰
2	Política de privacidad 2	100%	☰

Figura 3.43 Lista de políticas de privacidad lista para consolidar

id	politica_id	usuario_id	consolidar	finalizado
1	4	4	0	1
2	4	1	0	1
3	4	4	1	0
NULL	NULL	NULL	NULL	NULL

Figura 3.44 Registro de usuarios asignados para anotar "Política de privacidad 1"

3.1.2.1.6. HU-12: Cálculo de coeficiente inter-anotador

Como administrador de la herramienta requiero estar en la capacidad de:

1. En la página de visualización de políticas de privacidad por consolidar, tener una opción para ver el coeficiente inter-anotador y el número total de anotaciones que cada usuario realizó sobre la política. El coeficiente será calculado con el índice Alpha de Krippendorff.

Una vez que concluyó la etapa de anotación, como se muestra en la Figura 3.44, el backend de la herramienta realiza el cálculo del coeficiente inter-anotador. En la Figura 3.45 se presenta el resultado de este cálculo para las dos políticas del caso de prueba. Mientras tanto, en la Figura 3.46 se presenta el diálogo donde el coeficiente de "Política de

privacidad 1” es presentado. El cálculo manual del coeficiente inter-anotador para “Política de privacidad 2” se presenta en el ANEXO L.

id	nombre	url	fecha	asignada	coeficiente
4	Política de privacidad 1	https://www.lipsum.com/feed/html	2020-11-26	1	1
6	Política de privacidad 2	https://www.lipsum.com/feed/html	2020-11-26	1	-0.1

Figura 3.45 Registros de la base de datos de políticas de privacidad

Detalles de la política de privacidad		
Nombre: Política de privacidad 1		
Coeficiente interanotador = 1		
#	anotador	total anotaciones
1	felix.mino@epn.edu.ec	5
2	felix.minom@gmail.com	5

Figura 3.46 Detalles de "Política de privacidad 1"

3.1.2.1.7. HU-13: Consolidación de una política de privacidad

Los criterios de aceptación de esta historia de usuario junto con sus evidencias se encuentran en ANEXO K.

3.1.2.1.8. HU-14: Visualización, edición y eliminación de anotaciones en consolidación

Los criterios de aceptación de esta historia de usuario junto con sus evidencias se encuentran en el ANEXO K.

3.1.2.1.9. HU-15: Visualización y copia de anotaciones de la etapa de anotación en consolidación

Como administrador de la herramienta requiero:

1. Visualizar cada usuario que realizó la etapa de anotación con sus respectivas anotaciones, en cada sección de la política de privacidad.

En la Figura 3.47, se presenta parte de las anotaciones de la sección 1 de “Política de privacidad 1”.

2. Copiar una o más de las anotaciones realizadas por los usuarios en la etapa de anotación y que cuente como una anotación de consolidación.

En la Figura 3.48 se presenta la interfaz gráfica para la copia de una anotación. Y en la Figura 3.49 se presenta la anotación copiada.

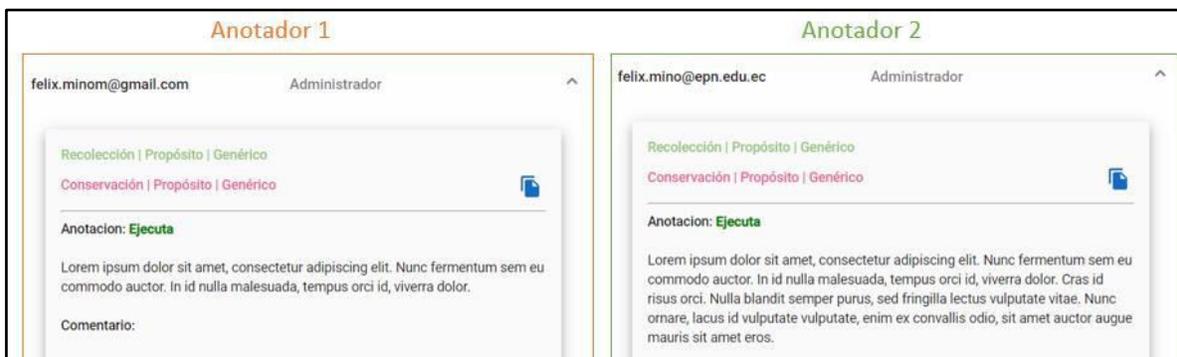


Figura 3.47 Resumen de anotaciones sección 1



Figura 3.48 Copia de anotación en etapa de consolidación

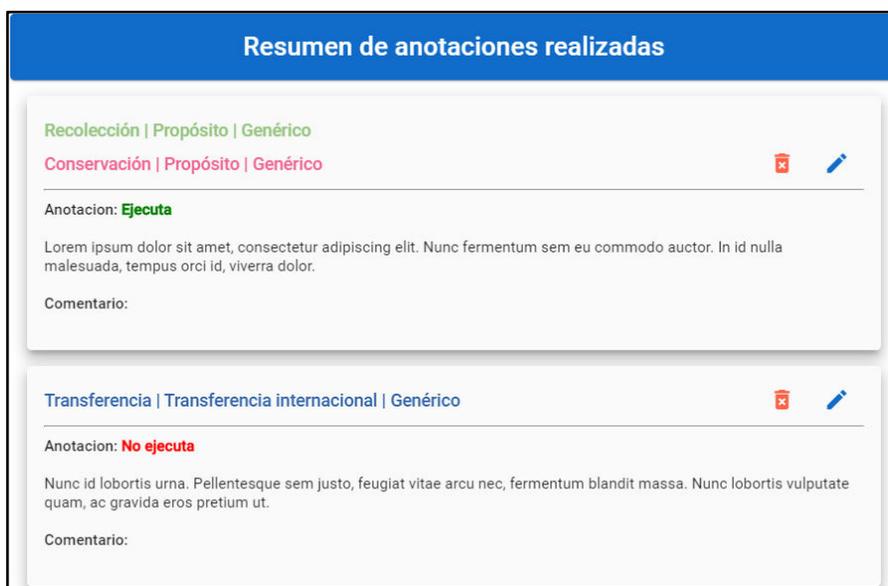


Figura 3.49 Resumen de anotaciones en etapa de consolidación

3.1.2.1.10. HU-16: Notificación de sección inconsistente en consolidación

Como administrador de la herramienta requiero:

1. Visualizar las secciones de una política con una notificación de consistencia. Se considera que una sección es consistente cuando todas las anotaciones de todos los usuarios son iguales, no se tomará en cuanto el texto y comentario. También se considera una sección consistente si no presenta ninguna anotación.

A continuación, se presenta los casos posibles de consistencia de una sección:

- Figura 3.50 una sección consistente sin anotaciones.
- Figura 3.51 una sección consistente con anotaciones iguales.
- Figura 3.52 una sección inconsistente.

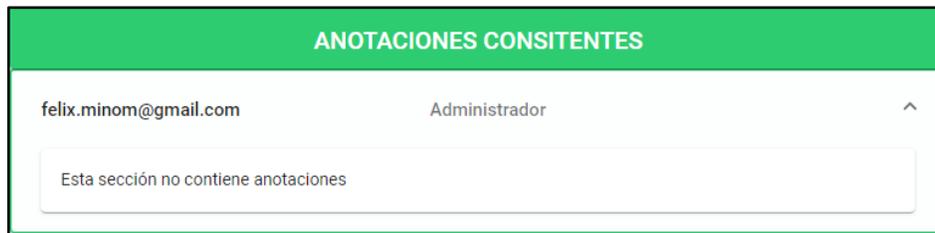


Figura 3.50 Sección 2 de "Política de privacidad 1", consistente sin anotaciones.

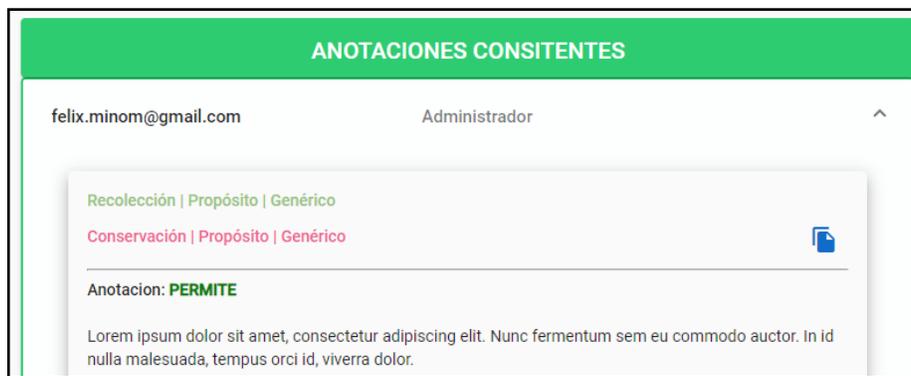


Figura 3.51 Sección 1 de "Política de privacidad 1", consistente con anotaciones.



Figura 3.52 Sección 3 de "Política de privacidad 1", inconsistente.

3.1.2.1.11. HU-17: *Página de bienvenida en herramienta de visualización*

Como usuario de la herramienta de visualización requiero:

1. Al ingresar a la herramienta de visualización tener una página de bienvenida que me da una visión general de este proyecto.

Explora nuestros resultados

Este proyecto pretende transparentar las políticas de privacidad y mostrar de manera más clara la manera en que tus datos son tratados. El gráfico que se encuentra a continuación hace un brevísimos resumen de la manera que esta herramienta funciona:

```
graph LR; A[Anotador] -- "Anota políticas de privacidad" --> H[Herramienta de anotación]; P[Política de privacidad] --> H; H -- "Proceso de anotación" --> PA[Política de privacidad anotada]; PA -- "Visualiza" --> U[Usuario final];
```

- Hemos escogido una serie de políticas de las paginas más populares para que estas sean anotadas de manera manual.
- Después de cumplir con un proceso de anotación, este es el resultado de los tratamientos que han sido encontrados en estas políticas.

[Explora nuestras políticas anotadas](#)

Figura 3.53 Página de bienvenida herramienta de visualización

3.1.2.1.12. HU-18: *Lista de políticas de privacidad para visualizar*

Como usuario de la herramienta de visualización requiero:

1. Ver las políticas que concluyeron el proceso de anotación en forma de lista. Una fila de una política de privacidad incluirá: índice, nombre, fecha en que se anotó y el número total de anotaciones.

Una vez concluido el caso de prueba ambas políticas de privacidad están disponibles para ser visualizadas. En la Figura 3.54 se evidencia y comprueba este criterio de aceptación.

#	Nombre de la política de privacidad	Fecha anotada	# anotaciones
1	Política de privacidad 1	2020-11-26	5
2	Política de privacidad 2	2020-11-26	4

Figura 3.54 Políticas de privacidad disponibles para visualización

2. Filtrar una política de privacidad por su nombre

En la Figura 3.55 se presenta las políticas de privacidad filtradas por el carácter “1”.

#	Nombre de la política de privacidad	Fecha anotada	# anotaciones
1	Política de privacidad 1	2020-11-26	5

Figura 3.55 Ejemplo de filtrado de lista de políticas de privacidad

3.1.2.1.13. HU-19: Visualización de una política de privacidad anotada

Como usuario de la herramienta de visualización requiero:

1. Al dar *click* en una política de la lista “Políticas de privacidad anotadas” ser redirigido a la página de visualización y observar resaltados los tratamientos de datos que fueron anotados sobre ésta.

El resultado de seleccionar “Política de privacidad 1” de la lista presentada en las pruebas anteriores se presenta en la Figura 3.56. Los tratamientos de datos se encuentran resaltados con el color del tratamiento correspondiente.

Categorías de tratamiento
Da click en una categoría para filtrar anotaciones

- Recolección (2)
- Conservación (3)
- Transferencia (2)
- Control del titular (2)

POLÍTICA DE PRIVACIDAD 1

Fecha en que se guardo la política: 2020-11-26

Sección 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc fermentum sem eu commodo auctor. In id nulla malesuada, tempus orci id, viverra dolor. Cras id risus orci. Nulla blandit semper purus, sed fringilla lectus vulputate vitae. Nunc ornare, lacus id vulputate vulputate, enim ex convallis odio, sit amet auctor augue mauris sit amet eros.

Nam maximus dolor in urna tristique molestie. Nunc id lobortis urna. Pellentesque sem justo, feugiat vitae arcu nec, fermentum blandit massa. Nunc lobortis vulputate quam, ac gravida eros pretium ut.

Nulla ex magna, sagittis sit amet dui et, venenatis viverra nulla. Nulla facilisi. Donec vulputate sit amet ipsum id tincidunt. Duis id ligula id nunc consectetur dictum. Praesent viverra odio purus, at blandit ipsum rhoncus ut. Donec suscipit neque sed rhoncus mollis. Sed sit amet urna at est ultricies volutpat ut nec massa. Sed sed ligula vel neque molestie consequat. Fusce sem diam, sagittis faucibus enim id, aliquet fringilla magna.

Nunc vel purus odio. Integer lacinia lorem dui, vel lobortis dui euismod eu. Integer ornare ante in urna dapibus malesuada.

Sección 2

Ut ullamcorper libero sit amet euismod pretium. Cras dictum erat nisi, in euismod ante vehicula ut. Ut sit amet cursus tellus. Curabitur hendrerit leo leo, non tempor enim tempor a. Vivamus quis dui vehicula, consequat dolor vel, vulputate mauris. Aenean sem arcu, gravida pulvinar fringilla ac, viverra in nibh. Nulla interdum, est sed lacinia sollicitudin, leo diam lobortis erat, et malesuada ligula leo sit amet ligula. Proin sagittis ligula et nulla sagittis, nec congue felis dignissim. Mauris diam neque, varius in magna sed, laoreet scelerisque odio. Duis nisi mi, scelerisque in porta non, dignissim sit amet tortor. Vestibulum fermentum nunc et urna pulvinar, nec luctus leo imperdiet. Suspendisse consectetur arcu sit amet dolor efficitur, ut vulputate ex pellentesque.

Sección 3

Donec ac metus molestie, vestibulum nibh a, euismod risus. Proin eu venenatis ipsum. In dignissim magna nec porttitor consequat. Fusce molestie lorem purus, eget tempus nisi iaculis sit amet. Phasellus sed rutrum sapien. Pellentesque quis dapibus turpis.

Nullam luctus vel erat ac efficitur. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce dapibus auctor dapibus. Phasellus rhoncus risus eget velit iaculis pellentesque. Mauris gravida id enim sit amet lacinia. Curabitur at maximus lectus.

Nulla lobortis iaculis erat. Vivamus nec tristique mi, eget blandit nunc. Phasellus volutpat nec tellus sed vehicula. In sit amet eros porttitor dui iaculis dianissim id vel lorem. Proin iaculis porta maana. vel blandit felis facilisis

Figura 3.56 Página de visualización de política de privacidad

2. Observar una anotación resaltada por el color del primer tratamiento de datos (en caso de que sea una anotación multivalor).

- Al poner el cursor sobre una anotación me muestre un cuadro de texto que contenga: el número de tratamientos que conforman la anotación, la acción (ejecuta o no ejecuta), una lista detalla (tratamiento, atributo y valor) para cada valor de la anotación y el comentario con el que fue anotado (en caso de existir).

Estos dos criterios de aceptación pueden comprobarse mediante la Figura 3.57. Como se observa en diálogo el primer tratamiento de datos es “Recolección | Propósito | Genérico” y su color es el que se muestra en el texto anotado. También se puede observar los detalles descritos en el criterio de aceptación tres.

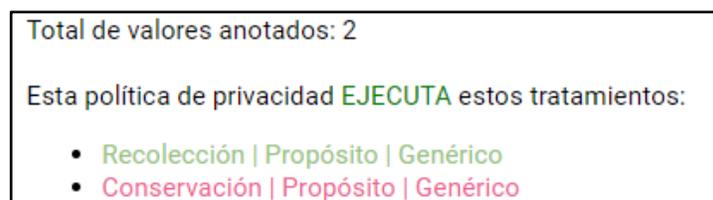


Figura 3.57 *Tooltip* con detalles de anotación

- Observar una lista de todos los tratamientos con los que se pudo anotar la política, es decir los tratamientos que tienen al menos un atributo y un valor.

Se puede comprobar mediante la Figura 3.56, donde se observa la lista de tratamientos.

- Observar junto a cada tratamiento el número de anotaciones que lo contienen y una barra del porcentaje que representa el uso de este tratamiento sobre el total de tratamientos presentes en la política.

Se puede comprobar mediante la Figura 3.56. Cada tratamiento de datos tiene

- Al dar *click* en un tratamiento de la lista expandir una lista de los atributos que contiene este y al dar *click* nuevamente colapsar los atributos.

En la Figura 3.58 se evidencia este comportamiento.

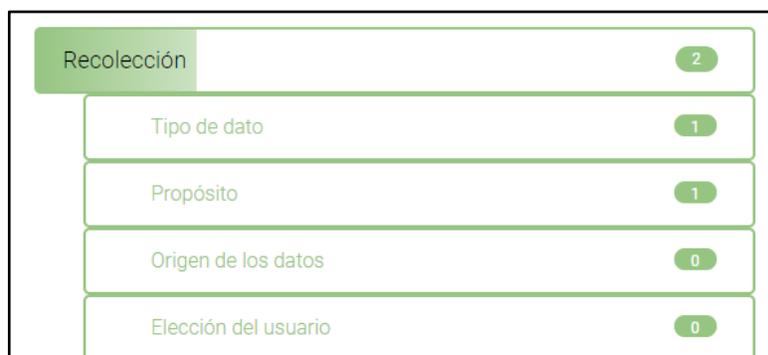


Figura 3.58 Tratamiento con atributos desplegados

3.1.2.1.14. HU-20: Filtrado de anotaciones en visualización

Como usuario de la herramienta de visualización requiero:

1. Al dar *click* sobre uno de los tratamientos listados en la página de visualización, la política de privacidad se filtre y muestre solamente las anotaciones que incluyan este tratamiento. Al dar *click* nuevamente sobre el tratamiento mostrar la política de privacidad sin filtros.

Este comportamiento se ilustra en la Figura 3.59. La política de privacidad es filtrada por el tratamiento “Recolección”, al dar *click* nuevamente sobre este tratamiento se obtiene la política de privacidad presentada en la Figura 3.56

2. Al dar *click* en un atributo de un tratamiento, filtrar la política de privacidad con las anotaciones que contengan ese atributo. Se mostrará un botón que permitirá limpiar el filtrado y mostrará la política de privacidad original.

En la Figura 3.60 se observa la política de privacidad filtrada por el tratamiento de datos “Recolección” y por su atributo “Tipo de dato” y al contrastar con la Tabla 3.3 se puede comprobar que este atributo existe solamente en la sección tres.

Categorías de tratamiento
Da click en una categoría para filtrar anotaciones
Filtrado por: Recolección

Recolección	2
Tipo de dato	1
Propósito	1
Origen de los datos	0
Elección del usuario	0
Conservación	3
Transferencia	2
Control del titular	2

POLÍTICA DE PRIVACIDAD 1

Fecha en que se guardo la política: 2020-11-26

Sección 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc fermentum sem eu commodo auctor. In id nulla malesuada, tempus orci id, viverra dolor. Cras id risus orci. Nulla blandit semper purus, sed fringilla lectus vulputate vitae. Nunc ornare, lacus id vulputate vulputate, enim ex convallis odio, sit amet auctor augue mauris sit amet eros.

Nam maximus dolor in urna tristique molestie. Nunc id lobortis urna. Pellentesque sem justo, feugiat vitae arcu nec, fermentum blandit massa. Nunc lobortis vulputate quam, ac gravida eros pretium ut.

Nulla ex magna, sagittis sit amet dui et, venenatis viverra nulla. Nulla facilisi. Donec vulputate sit amet ipsum id tincidunt. Duis id ligula id nunc consectetur dictum. Praesent viverra odio purus, at blandit ipsum rhoncus ut. Donec suscipit neque sed rhoncus mollis. Sed sit amet urna at est ultricies volutpat ut nec massa. Sed sed ligula vel neque molestie consequat. Fusce sem diam, sagittis faucibus enim id, aliquet fringilla magna. Nunc vel purus odio. Integer lacinia lorem dui, vel lobortis dui euismod eu. Integer ornare ante in urna dapibus malesuada.

Sección 2

Ut ullamcorper libero sit amet euismod pretium. Cras dictum erat nisi, in euismod ante vehicula ut. Ut sit amet cursus tellus. Curabitur hendrerit leo leo, non tempor enim tempor a. Vivamus quis dui vehicula, consequat dolor vel, vulputate mauris. Aenean sem arcu, gravida pulvinar fringilla ac, viverra in nibh. Nulla interdum, est sed lacinia sollicitudin, leo diam lobortis erat, et malesuada ligula leo sit amet ligula. Proin sagittis ligula et nulla sagittis, nec congue felis dignissim. Mauris diam neque, varius in magna sed, laoreet scelerisque odio. Duis nisi mi, scelerisque in porta non, dignissim sit amet tortor. Vestibulum fermentum nunc et urna pulvinar, nec luctus leo imperdiet. Suspendisse consectetur arcu sit amet dolor efficitur, ut vulputate ex pellentesque.

Sección 3

Donec ac metus molestie, vestibulum nibh a, euismod risus. Proin eu venenatis ipsum. In dignissim magna nec porttitor consequat. Fusce molestie lorem purus, eget tempus nisi iaculis sit amet. Phasellus sed rutrum sapien. Pellentesque quis dapibus turpis.

Nullam luctus vel erat ac efficitur. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce dapibus auctor dapibus. Phasellus rhoncus risus eget velit iaculis pellentesque. Mauris gravida id enim sit amet lacinia. Curabitur at maximus lectus.

Nulla lobortis iaculis erat. Vivamus nec tristique mi, eget blandit nunc. Phasellus volutpat nec tellus sed vehicula. In sit amet eros porttitor dui iaculis dignissim id vel lorem. Proin iaculis porta magna, vel blandit felis facilisis elementum. Nulla sodales tincidunt aliquet. Integer aliquam dolor ut lectus rutrum consectetur. Cras eget maximus mi.

Figura 3.59 Anotaciones de filtradas por tratamiento "Recolección"

Categorías de tratamiento
Da click en una categoría para filtrar anotaciones

Filtrado por: Recolección | Tipo de dato

Recolección	2
Tipo de dato	1
Propósito	1
Origen de los datos	0
Elección del usuario	0
Conservación	3
Transferencia	2
Control del titular	2

POLÍTICA DE PRIVACIDAD 1

Fecha en que se guardo la política: 2020-11-26

Sección 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc fermentum sem eu commodo auctor. In id nulla malesuada, tempus orci id, viverra dolor. Cras id risus orci. Nulla blandit semper purus, sed fringilla lectus vulputate vitae. Nunc ornare, lacus id vulputate vulputate, enim ex convallis odio, sit amet auctor augue mauris sit amet eros. Nam maximus dolor in urna tristique molestie. Nunc id lobortis urna. Pellentesque sem justo, feugiat vitae arcu nec, fermentum blandit massa. Nunc lobortis vulputate quam, ac gravida eros pretium ut.

Nulla ex magna, sagittis sit amet dui et, venenatis viverra nulla. Nulla facilisi. Donec vulputate sit amet ipsum id tincidunt. Duis id ligula id nunc consectetur dictum. Praesent viverra odio purus, at blandit ipsum rhoncus ut. Donec suscipit neque sed rhoncus mollis. Sed sit amet urna at est ultricies volutpat ut nec massa. Sed sed ligula vel neque molestie consequat. Fusce sem diam, sagittis faucibus enim id, aliquet fringilla magna. Nunc vel purus odio. Integer lacinia lorem dui, vel lobortis dui euismod eu. Integer ornare ante in urna dapibus malesuada.

Sección 2

Ut ullamcorper libero sit amet euismod pretium. Cras dictum erat nisi, in euismod ante vehicula ut. Ut sit amet cursus tellus. Curabitur hendrerit leo leo, non tempor enim tempor a. Vivamus quis dui vehicula, consequat dolor vel, vulputate mauris. Aenean sem arcu, gravida pulvinar fringilla ac, viverra in nibh. Nulla interdum, est sed lacinia sollicitudin, leo diam lobortis erat, et malesuada ligula leo sit amet ligula. Proin sagittis ligula et nulla sagittis, nec congue felis dignissim. Mauris diam neque, varius in magna sed, laoreet scelerisque odio. Duis nisi mi, scelerisque in porta non, dignissim sit amet tortor. Vestibulum fermentum nunc et urna pulvinar, nec luctus leo imperdiet. Suspendisse consectetur arcu sit amet dolor efficitur, ut vulputate ex pellentesque.

Sección 3

Donec ac metus molestie, vestibulum nibh a, euismod risus. Proin eu venenatis ipsum. In dignissim magna nec porttitor consequat. Fusce molestie lorem purus, eget tempus nisi iaculis sit amet. Phasellus sed rutrum sapien. Pellentesque quis dapibus turpis.

Nullam luctus vel erat ac efficitur. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce dapibus auctor dapibus. Phasellus rhoncus risus eget velit iaculis pellentesque. Mauris gravida id enim sit amet lacinia. Curabitur at maximus lectus.

Nulla lobortis iaculis erat. Vivamus nec tristique mi, eget blandit nunc. Phasellus volutpat nec tellus sed vehicula. In sit amet eros porttitor dui iaculis dignissim id vel lorem. Proin iaculis porta magna, vel blandit felis facilisis elementum. Nulla sodales tincidunt aliquet. Integer aliquam dolor ut lectus rutrum consectetur. Cras eget maximus mi.

Figura 3.60 Anotaciones filtradas por tratamiento "Recolección" y atributo "Tipo de dato"

3.2. VALIDACIÓN

En esta sección se presenta la evidencia con la que se validó el funcionamiento de la herramienta: la validación general de la herramienta a través de la anotación de cinco políticas de privacidad, una encuesta de satisfacción de los requisitos funcionales, una encuesta de satisfacción de requisitos no funcionales y el uso de la herramienta en procesos de investigación

3.2.1. FUNCIONAMIENTO GENERAL DE LA HERRAMIENTA

Una vez concluidas las pruebas de funcionamiento se realizó la anotación de cinco políticas de privacidad. Tres políticas de privacidad son de aplicaciones desarrolladas por instituciones ecuatorianas, mientras que dos pertenecen a aplicaciones mundialmente conocidas. En la Figura 3.61 se presentan las aplicaciones elegidas para esta etapa.

El objetivo de esta etapa es que el *product owner* pueda validar el funcionamiento general de la herramienta a través de la anotación y consolidación de políticas de privacidad reales. Así como contrastar los resultados obtenidos de las políticas de privacidad de aplicaciones ecuatorianas con los resultados de las aplicaciones de plataformas mundiales.

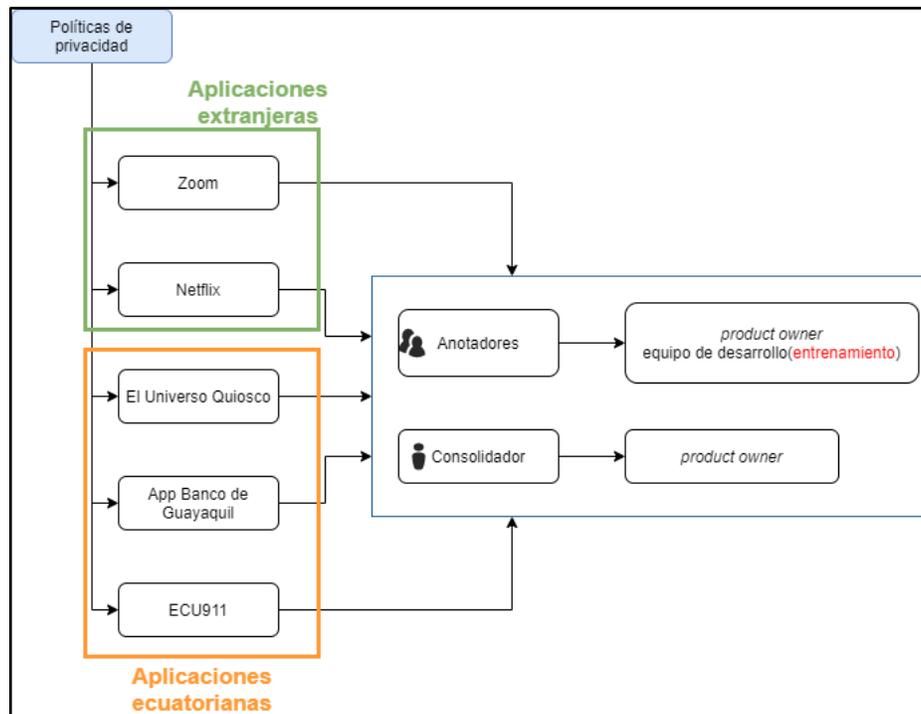


Figura 3.61 Resumen de políticas de privacidad para fase de validación

3.2.2. ENCUESTA DE SATISFACCIÓN REQUISITOS FUNCIONALES

Así también, con el objetivo de validar las funcionalidades de la herramienta y en búsqueda de sugerencias para futuros trabajos de titulación, se realizó una encuesta de satisfacción a través de Google Forms⁴². La encuesta cuenta de 4 secciones:

- Preguntas de validación de historias de usuario de herramienta de anotación.
- Preguntas de validación de historias de usuario de herramienta de visualización.
- Preguntas generales (aplican a herramienta de anotación y visualización).
- Preguntas de opinión.

Esta encuesta se realizó al *product owner* y a otro usuario que hizo uso de la herramienta en un proceso de experimentación. Las preguntas se encuentran en el ANEXO M.

3.2.3. ENCUESTA DE SATISFACCIÓN REQUISITOS NO FUNCIONALES

Por último, se realizó una encuesta de satisfacción al *product owner* en cuanto a los requisitos no funcionales de la herramienta. Las preguntas de esta encuesta se presentan en la Tabla 3.4.

⁴² **Google Forms:** es un software de administración de encuestas que se incluye en el paquete de software Google Docs. Permite recopilar información de los usuarios a través de encuestas.

Tabla 3.4 Preguntas de validación requerimientos no funcionales

Pregunta
¿La herramienta soporta la anotación concurrente de políticas de privacidad?
¿La herramienta de anotación es adaptable en el sentido que permite la configuración de cualquier tipo de tratamiento bajo la estructura de tres niveles (tratamiento, atributos, y valores)?
¿Considera que la herramienta es fácilmente mantenible (capacidad de modificación)?

Las respuestas de la encuesta junto a su análisis y casos de prueba que se realizaron con el fin de validar estos requerimientos se presentan en la sección 3.3.

3.2.4. USO DE LA HERRAMIENTA EN PROCESOS DE INVESTIGACIÓN

Además de los procesos de validación presentados en las secciones anteriores, el *product owner* realizó la anotación de 100 políticas de privacidad en inglés demostrando la flexibilidad de la herramienta en cuanto al lenguaje de las políticas de privacidad. En el siguiente enlace se puede explorar los resultados de las anotaciones <https://github.com/PrivApp/IT100-Corpus>.

3.3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados y discusión de: la anotación de las 5 políticas de privacidad explicadas en la sección 3.2.1, la encuesta de satisfacción de requisitos funcionales y la encuesta de satisfacción de requisitos no funcionales

3.3.1. FUNCIONAMIENTO GENERAL DE LA HERRAMIENTA

Una vez concluida la anotación de las cinco políticas de privacidad mencionadas previamente se obtuvo los siguientes resultados.

Por sugerencia del *product owner* se decidió eliminar el atributo “Elección de usuario” del tratamiento “Recolección”, durante la anotación de las 5 políticas de privacidad ninguna anotación presentó este atributo. Por lo tanto, la lista final de tratamientos, atributos y valores, en principio obtenida del PLODP y luego depurada por el *product owner*, se presenta en la Tabla 3.5.

Tabla 3.5 Lista final de tratamientos, atributos y valores

Tratamiento de datos	Atributo	Valor
Recolección	Tipo de dato	Genérico
	Propósito	Genérico
	Origen de los datos	Genérico
Conservación	Periodo de conservación	Genérico
	Propósito	Genérico
Transferencia	Transferencia internacional	Genérico
	Transferencia a terceros	Genérico
	Propósito	Genérico
Control del titular	Acceso	Genérico
	Edición	Genérico
	Eliminación	Genérico

En cuanto a la extensión de las políticas de las políticas de privacidad existe una clara diferencia entre las políticas de privacidad de las aplicaciones extranjeras y las políticas de privacidad ecuatorianas. En promedio las aplicaciones extranjeras presentan 33,5 secciones por política de privacidad, mientras que las aplicaciones ecuatorianas tienen una extensión de 5,67 secciones. Esto se resume en la Figura 3.62.

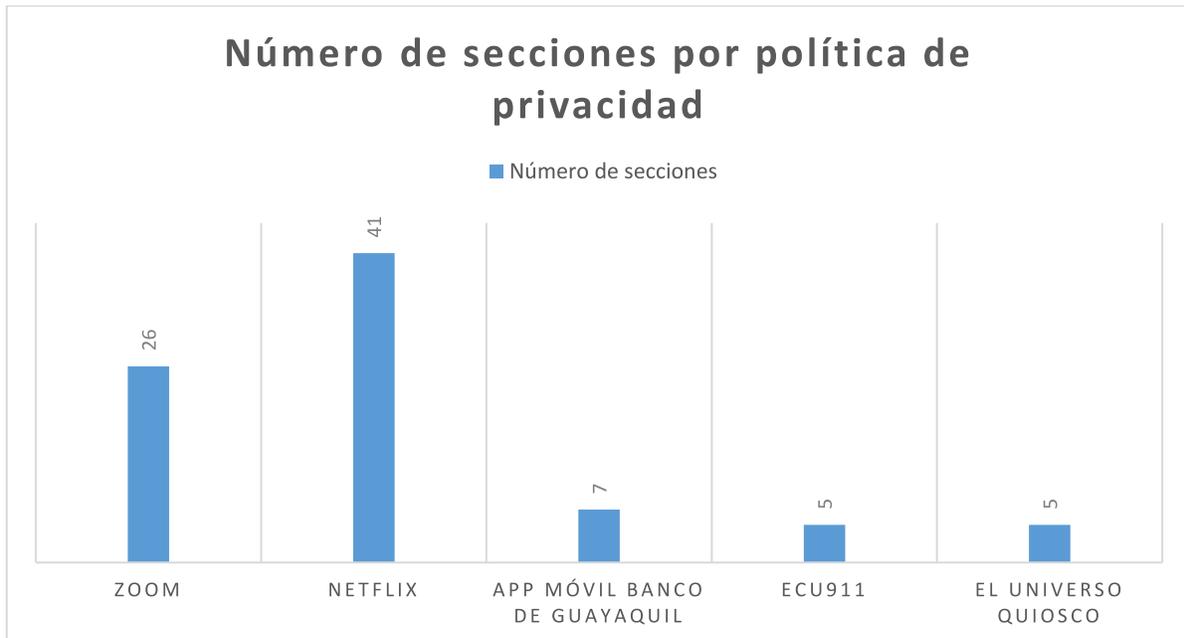


Figura 3.62 Secciones totales por política de privacidad

Así mismo, esta diferencia se presenta en el número de palabras que tiene cada política de privacidad, lo cual se resume en la Figura 3.63.

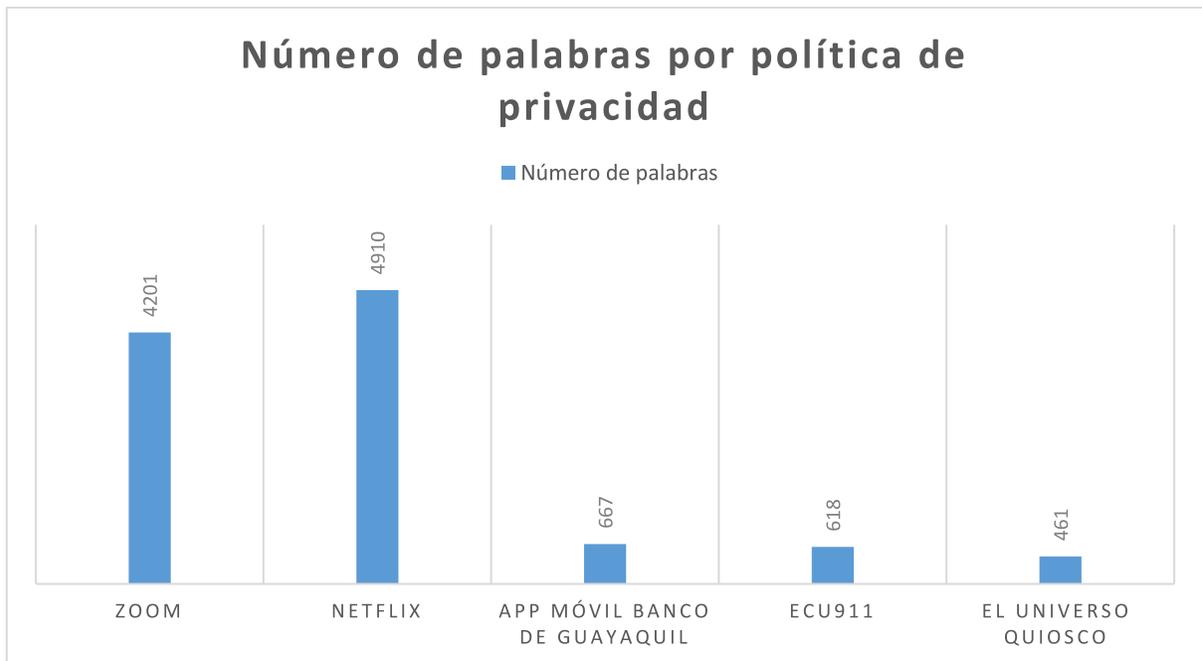


Figura 3.63 Número de palabras por política de privacidad

La extensión de una política de privacidad tiene un impacto directo sobre el número de anotaciones que se realizan sobre ésta. En la Figura 3.64 se presenta el número de anotaciones por política de privacidad en la etapa de consolidación. En promedio sobre las políticas de privacidad extranjeras se realizaron 52 anotaciones, mientras que en las ecuatorianas 3,67 anotaciones.

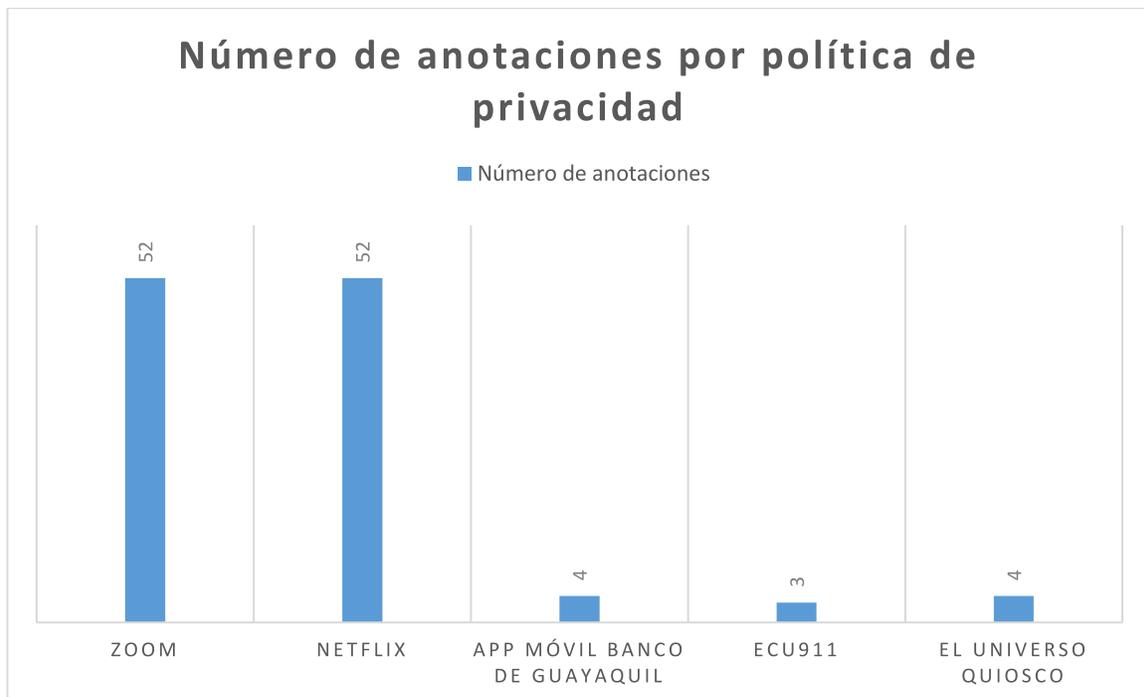


Figura 3.64 Total de anotaciones por política de privacidad

Así también, la extensión de una política de privacidad tiene relación directa con el nivel de transparencia de sus tratamientos de datos. Por ejemplo, mientras que la política de privacidad de Zoom presenta 9 secciones, con un total de 1046 palabras, relacionadas a la recolección de datos personales, la política de privacidad de la aplicación El Universo Quiosco presenta el siguiente segmento de texto respecto al mismo tratamiento: *“Está bajo las normas de confidencialidad y/o privacidad toda aquella información que se obtiene del usuario al utilizar los servicios que ofrecemos mediante nuestra aplicación o mientras visite los servicios que ofrecemos a través de la misma”*. Como es evidente en la política de privacidad de la aplicación ecuatoriana en ningún momento se detalla la información que se recolecta del usuario.

Respecto a los tratamientos de datos, en la Figura 3.65 se presenta el número de anotaciones que se realizó por cada tratamiento de datos. Es evidente que el tratamiento de datos que más se ejecuta es la recolección de datos personales, seguido por la transferencia de datos, control del titular y finalmente la conservación de datos.

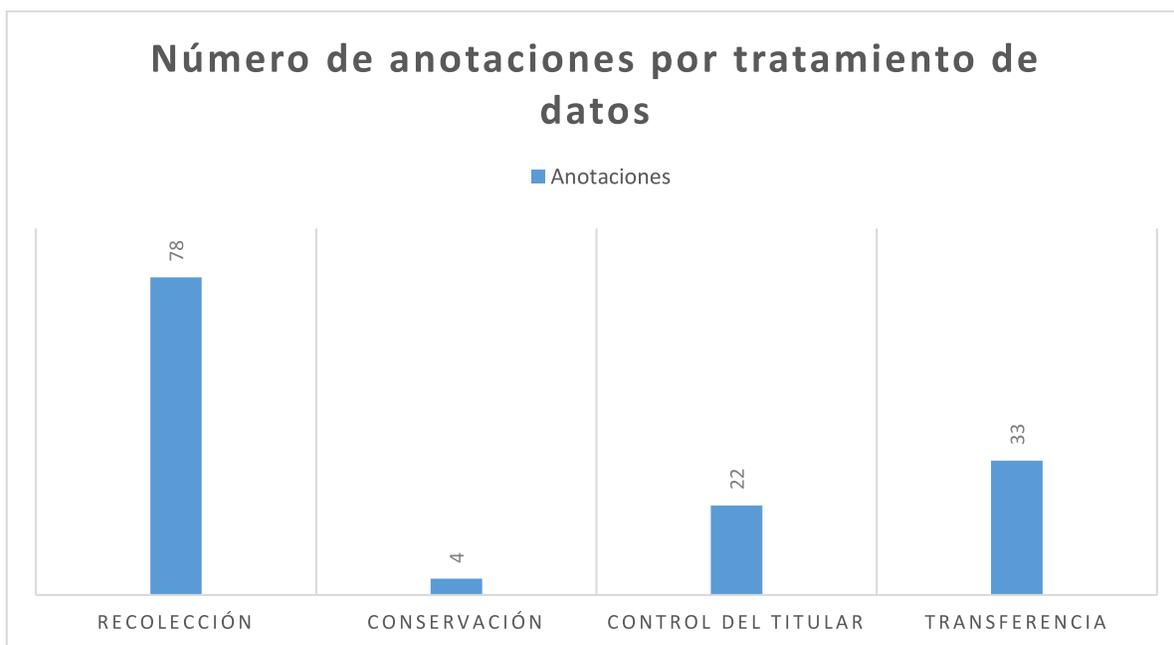


Figura 3.65 Número de anotaciones por tratamiento de datos

En la Figura 3.66 se presenta el número de anotaciones que se realizó por cada tratamiento de datos sobre cada una de las políticas de privacidad anotadas en la sección de validación. Donde se mantiene la tendencia donde la recolección de datos personales es el tratamiento de datos más ejecutado.

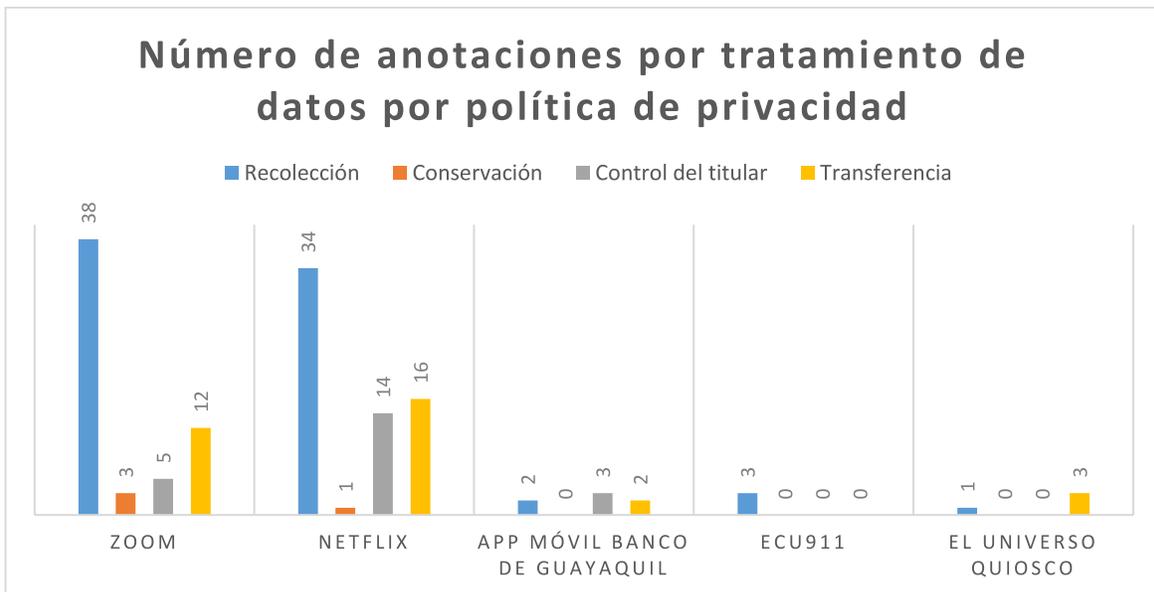


Figura 3.66 Número de anotaciones por tratamiento de datos por política de privacidad

En cuanto a la acción que realizan las anotaciones. Es decir si ejecutan o no los tratamientos de datos que contienen se obtuvo: 130 anotaciones ejecutan los tratamientos que mencionan, mientras que 7 anotaciones no los ejecutan.

Por último, cabe mencionar que la anotación de las 5 políticas de privacidad de esta sección se realizó como mecanismo de validación del funcionamiento de las herramientas de anotación y visualización. La anotación de estas políticas está lejos de brindar resultados de prevalencia de los tratamientos de datos que una política de privacidad declara, esto es tema para un trabajo futuro el cual se detalla en la sección de conclusiones y recomendaciones.

3.3.2. ENCUESTA DE SATISFACCIÓN REQUISITOS FUNCIONALES

Al realizar la encuesta de satisfacción a dos personas: *product owner* y una persona que está usando la herramienta como soporte para un trabajo de investigación, se obtuvo los resultados que se presentan en la Tabla 3.6. Las respuestas completas se encuentran en el ANEXO M. Cabe aclarar que ambos usuarios tenían un rol de administrador en la herramienta de anotación.

Tabla 3.6 Resultados encuesta de satisfacción de requisitos funcionales

Encuesta de satisfacción herramientas de anotación y visualización
La información proporcionada en esta encuesta será usada para la validación del proyecto: "DESARROLLO DE UNA HERRAMIENTA WEB PARA LA ANOTACIÓN DE TRATAMIENTOS DE DATOS PERSONALES EN POLÍTICAS DE PRIVACIDAD EN ESPAÑOL"

Preguntas herramienta de anotación	Si	No	Parcialmente
P1: ¿Pudo iniciar y cerrar sesión?	2	0	0
P2: ¿Pudo crear, visualizar, editar y eliminar los usuarios de la herramienta?	2	0	0
P3: ¿Pudo crear, visualizar, editar y eliminar los tratamientos de datos?	2	0	0
P4: ¿Pudo crear, visualizar, editar y eliminar los atributos de un tratamiento de datos?	2	0	0
P5: ¿Pudo crear, visualizar, editar y eliminar los valores de un atributo?	2	0	0
P6: ¿Pudo crear, visualizar, editar y eliminar las políticas de privacidad?	1	0	1
P7: ¿Pudo visualizar las políticas de privacidad que le han sido asignadas para la etapa de anotación?	2	0	0
P8: ¿Pudo anotar los tratamientos, atributos y valores de una política de privacidad?	2	0	0
P9: ¿Pudo visualizar, editar y eliminar las anotaciones realizadas en la etapa de anotación?	2	0	0
P10: Al estar en entrenamiento, ¿Fue notificado de inconsistencias en las anotaciones?	1	1*	0
P11: ¿Pudo visualizar las políticas de privacidad que le han sido asignadas para consolidar?	2	0	0
P12: En la etapa de consolidación ¿Pudo observar el coeficiente interanotador de una política de privacidad?	2	0	0
P13: ¿Pudo consolidar los tratamientos, atributos y valores de una política de privacidad?	2	0	0
P14: ¿Pudo visualizar, editar y eliminar las anotaciones realizadas en la etapa de consolidación?	2	0	0
P15: En la etapa de consolidación ¿Pudo visualizar las anotaciones realizadas en la etapa de anotación por los distintos usuarios (anotadores y administradores)?	2	0	0
P16: En la etapa de consolidación ¿Fue notificado cuando una sección era inconsistente?	2	0	0
Preguntas herramienta de visualización	Si	No	Parcialmente
P17: ¿Pudo observar la página de bienvenida?	2	0	0
P18: ¿Pudo observar la lista de políticas de privacidad que han sido anotadas?	2	0	0
P19: Al seleccionar una política de privacidad de la lista anterior ¿Pudo observar las anotaciones resaltadas?	2	0	0
P20: ¿Pudo filtrar las anotaciones de una política de privacidad por el tratamiento de datos o por alguno de sus atributos?	2	0	0

Preguntas generales	Respuesta
Responda en una escala del 1-5 donde 1 es 'En desacuerdo' y 5 'Totalmente de acuerdo' **	
PG1: ¿Las herramientas de anotación y visualización son fiables en cuanto a los datos que presentan y almacenan?	5
PG2: ¿Las interfaces de usuario son lo suficientemente intuitivas como para usar las herramientas sin necesidad de que una persona le guíe en su manejo?	3,5
PG3: ¿Considera que las herramientas podrían usarse en experimentos reales?	5
Responda en una escala del 1-5 donde 1 es 'No cumple las expectativas' y 5 'Cumple las expectativas' **	
PG4: De manera general ¿Cómo calificaría a la herramienta de anotación?	4,5
PG5: De manera general ¿Cómo calificaría a la herramienta de visualización?	4,5
Preguntas de opinión	
PO1: ¿Qué mejoras considera que serían útiles una vez implementadas en las herramientas de anotación y visualización?	
<p><i>Product owner:</i></p> <p>Herramienta de anotación:</p> <ul style="list-style-type: none"> - Permitir la configuración de atributos con la etiqueta de obligatorio y opcional. - Permitir el cálculo del alpha a nivel de segmento. - Carga automática de textos de políticas a partir de un enlace (URL). - Consolidación automática de segmentos de políticas consistentes. <p>Herramienta de visualización:</p> <ul style="list-style-type: none"> - Permitir la exportación de las anotaciones en formatos estructurados como YAML. - Permitir el filtrado a nivel de valor. <p>Usuario:</p> <ul style="list-style-type: none"> - La adaptación a cualquier tamaño del navegador. - Un lugar en el que cada anotador pueda ver las anotaciones que ha hecho de una política aunque no sean las que se han aprobado en consolidación. 	
<p>Observaciones:</p> <p>*El usuario que respondió "No" a la pregunta P10 nunca estuvo en entrenamiento.</p> <p>**En esta sección se presenta un promedio de las respuestas de los usuarios.</p>	

Se validó que las historias de usuario fueran implementadas de manera satisfactoria, ya que cada pregunta desde P1 hasta P20 corresponde a una historia de usuario. A excepción de las preguntas P6 y P10 el resto de las preguntas fueron respondidas de manera positiva. La pregunta 6 que corresponde a la administración de políticas de privacidad, al preguntar al usuario que respondió "Parcialmente" supo manifestar que en ocasiones es necesario eliminar una política de privacidad a pesar de que se realizaron anotaciones sobre esta, esto se considerará para la sección de recomendaciones. Mientras que la pregunta P10

tiene una respuesta negativa ya que este usuario nunca pasó por la etapa de entrenamiento.

En cuanto a las preguntas generales, se observa que la calificación más baja fue obtenida por la pregunta PG2 que trata sobre la intuición de un usuario sobre de las interfaces visuales. Este aspecto puede ser mejorado en trabajos futuros junto al *product owner*.

La sección de preguntas de opinión será considerada en el capítulo 4. Cada mejora propuesta será considerada una recomendación.

3.3.3. ENCUESTA DE SATISFACCIÓN REQUISITOS NO FUNCIONALES

Las respuestas del *product owner* sobre esta encuesta se presentan en la Tabla 3.7.

Tabla 3.7 Encuesta de satisfacción de requisitos no funcionales respondida

Pregunta	Respuesta
¿La herramienta soporta la anotación concurrente de políticas de privacidad?	Si
¿La herramienta de anotación es adaptable en el sentido que permite la configuración de cualquier tipo de tratamiento bajo la estructura de tres niveles (tratamiento, atributos, y valores)?	Si
¿Considera que la herramienta es fácilmente mantenible (capacidad de modificación)?	Si

Además de la encuesta, el equipo de desarrollo realizó casos de prueba donde varios ordenadores conectados de manera simultánea hacían uso de las herramientas de anotación y visualización, a lo largo de las pruebas no se encontraron inconvenientes con el uso de las herramientas.

Por otro lado, el *product owner* configuró distintos tratamientos de datos con sus atributos y valores en un proceso de investigación. En total creó 2 tratamientos, 6 atributos y 63 valores, demostrando la flexibilidad de la herramienta de anotación.

Con esta evidencia se demuestra que los requerimientos no funcionales fueron alcanzados durante el desarrollo de la herramienta de anotación y visualización.

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones y recomendaciones establecidas con el desarrollo de este proyecto de titulación.

4.1. CONCLUSIONES

- Los fundamentos teóricos investigados para la realización de este trabajo de titulación permitieron: 1) entender el contexto en el cual se iba a desarrollar la herramienta y la necesidad de su creación. Estos conceptos incluyen: privacidad, datos personales, tratamiento de datos personales y herramientas de anotación de tratamientos de datos personales expresados en políticas de privacidad. 2) guiar el desarrollo e implementación de la herramienta. Algunos de estos elementos son: la metodología ágil Scrum, los lenguajes de programación Python y TypeScript y los *frameworks* Flask y Angular.
- Los módulos de la herramienta fueron diseñados con la ayuda de la metodología Scrum. Al usar una metodología de desarrollo ágil se tuvo ventajas como: constante interacción entre todo el equipo Scrum, desarrollo iterativo de la herramienta y flexibilidad en la planificación de los módulos.
- Finalizado el presente trabajo se obtuvo una herramienta web que asiste en el proceso de anotación de tratamientos de datos expresados en políticas de privacidad en español. Algunas características relevantes de la herramienta son: la capacidad que tiene el administrador de la herramienta de gestionar todos los elementos que intervienen en la anotación de una política de privacidad, permite la anotación simultánea sobre una misma política de privacidad o diferentes políticas, previene al usuario anotador (en entrenamiento) de posibles errores en los tratamientos de datos elegidos al momento de generar una anotación, provee de un coeficiente inter-anotador el cual permite identificar de manera sencilla el nivel de fiabilidad que un consolidador puede tener sobre las anotaciones, y provee una herramienta de visualización que permite presentar de manera gráfica las anotaciones realizadas en la etapa de consolidación. Una de las limitantes de la herramienta es que permite solamente la anotación de políticas de privacidad bajo una jurisdicción, esto se consideró como una mejora a la herramienta y se encuentra en la sección de recomendaciones.

- Mediante las pruebas de funcionalidad y las encuestas de satisfacción se verificó que la herramienta cumple con los requerimientos funcionales y no funcionales planteados desde el inicio por el *product owner*.
- Hay que destacar que la herramienta fue usada en investigaciones reales, lo que resalta su fiabilidad en cuanto a los datos que presenta y almacena. La herramienta fue usada para llevar a cabo una investigación en el contexto de las transferencias internacionales en base al RDGP, cuyo corpus es públicamente accesible en <https://github.com/PrivApp/IT100-Corpus>.
- Los resultados de la anotación de las políticas de privacidad de organizaciones extranjeras y ecuatorianas sugieren una falta de transparencia en las segundas. Aunque la muestra considerada (n=5) está lejos de ser representativa, creemos que, al no existir una ley de protección de datos vigente, los resultados no se alejan de la realidad. La herramienta desarrollada puede servir como base para generar un corpus y a partir de allí modelos de clasificación automática que permitan realizar un estudio de prevalencia sobre una muestra significativa.
- En base a la retroalimentación recibida a través de la encuesta de satisfacción de requerimientos funcionales, reconocemos que la herramienta requiere funcionalidad adicional que ha surgido durante el uso de la herramienta en escenarios reales de experimentación. En este sentido, durante el desarrollo de la herramienta se tuvo presente buenas prácticas de programación, incluyendo la legibilidad del código, reutilización de funciones y organización modular del código. De este modo, la herramienta puede ser fácilmente extendida por terceras personas que requieran implementar nuevas funcionalidades.
- La anotación o consolidación de los tratamientos de datos sobre una política de privacidad es de naturaleza cualitativa e interpretativa. Dado que el proceso es realizado por seres humanos, podría haber un sesgo al momento de decidir anotar o no un tratamiento de datos y sus elementos, que, en última instancia se convierte en una amenaza a la validez de los resultados. En la herramienta existen medidas que mitigan esta amenaza, una de ellas es que se requiere un mínimo de dos anotadores sobre una política de privacidad, que luego es consolidada. En la etapa de anotación es necesario medir el acuerdo que existe entre los usuarios que realizan este proceso, por lo que la medición de este acuerdo se automatizó a través del cálculo del coeficiente Alpha de Krippendorff, llamado coeficiente inter-anotador en este trabajo.

4.2. RECOMENDACIONES

Las recomendaciones que se presentan en esta sección fueron recopiladas de la encuesta de satisfacción de requisitos funcionales y del criterio del equipo de desarrollo.

4.2.1. RECOMENDACIONES OBTENIDAS DE LA ENCUESTA DE SATISFACCIÓN DE REQUISITOS FUNCIONALES

4.2.1.1. Herramienta de anotación

- La herramienta de anotación debería permitir marcar un atributo con la etiqueta obligatorio u opcional, de esta manera se tiene una mejor visión de qué atributos son requeridos en una política de privacidad.
- La herramienta de anotación debería permitir configurar si el cálculo del coeficiente inter-anotador se realiza a nivel de política de privacidad o de sus secciones.
- Con el propósito de mejorar la legibilidad de las políticas de privacidad y mantener algunas de sus características, como: listas numeradas y no numeradas, texto en negrita y cursiva, distinción entre títulos y subtítulos, entre otros, se recomienda leer el texto de una política en formato `markdown` que luego pueden ser transformadas a etiquetas `html`. Un siguiente paso sería la descarga automática de una política de privacidad a través de *web scraping*⁴³ a partir de su enlace (URL).
- En la etapa de consolidación, si una sección es consistente la herramienta debería consolidar esta sección automáticamente, es decir copiar las anotaciones de la sección de la etapa de anotación hacia la etapa de consolidación.
- En la herramienta de anotación debería existir un módulo que permita visualizar las anotaciones que cada anotador realizó aunque no sean las que se aprobaron en consolidación.
- Se podrían presentar las anotaciones en la etapa de anotación y consolidación de manera similar a la que se presenta en la herramienta de visualización, es decir resaltadas por el color del primer tratamiento de datos.

4.2.1.2. Herramienta de visualización

- La herramienta de visualización debería permitir la exportación de las anotaciones de una o varias políticas de privacidad hacia un archivo de formato estructurado,

⁴³ **Web scraping:** o raspado web, es una técnica utilizada mediante programas de software para extraer información de sitios web.

por ejemplo YAML. Por el momento este proceso se realiza de manera manual haciendo consultas directamente en la base de datos.

- La herramienta de visualización debería permitir el filtrado de las anotaciones a nivel de valor.
- Las herramientas de anotación y visualización deberían ser adaptables a distintas resoluciones y navegadores. Se recomienda las siguientes resoluciones: HD, FHD y UHD y los siguientes navegadores: Google Chrome, Firefox y Safari.

4.2.2. RECOMENDACIONES DEL EQUIPO DE DESARROLLO

- Al momento, la herramienta está limitada a trabajar bajo una jurisdicción. Por lo tanto, se recomienda que:
 1. La herramienta de anotación permita la anotación de políticas de privacidad de distintas jurisdicciones, por ejemplo: RGPD y PLODP. Es decir que un conjunto de tratamientos, atributos y valores se relacionen a una jurisdicción.
 2. La herramienta de visualización permita filtrar las políticas de privacidad anotadas bajo cierta jurisdicción.
- Como se mencionó en la sección de implementación, el uso de un sistema de tipado previene errores en tiempo de compilación. Por esto, se recomienda implementar la librería `mypy` para realizar chequeo estático de tipos en el lenguaje de programación Python, en el *backend* de la aplicación.
- Por ahora las herramientas de anotación y visualización no cuentan con ningún tipo de pruebas automatizadas. Se recomienda implementar pruebas unitarias, de integración y `e2e`⁴⁴. De esta manera, al momento de realizar modificaciones sobre el código actual o de implementar nuevas funcionalidades se puede asegurar que la herramienta no sea afectada en alguno de sus módulos.
- Con el propósito de optimizar el proceso de despliegue de una nueva versión de las herramientas de anotación y visualización y el *backend* de la herramienta, se recomienda implementar un ciclo de integración continua para que el despliegue se realice de manera automática.

⁴⁴ **Pruebas e2e:** simulan el comportamiento de un usuario real. Prueban toda la aplicación de principio a fin, cubriendo así secciones que las pruebas unitarias y las pruebas de integración no cubren.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] ENISA, *Privacy and data protection in mobile applications: A study on the app development ecosystem and the technical implementation of GDPR*, núm. November. 2017.
- [2] BBC, “5 claves para entender el escándalo de Cambridge Analytica que hizo que Facebook perdiera US\$37.000 millones en un día - BBC News Mundo”, 2018. [En línea]. Disponible en: <https://www.bbc.com/mundo/noticias-43472797>. [Consultado: 22-oct-2019].
- [3] Unión Europea, “Reglamento General de Protección de Datos”, *D. Of. la Unión Eur.*, vol. 2014, núm. 119, pp. 1–88, 2016.
- [4] República del Ecuador, “Proyecto de Ley Orgánica de Protección de Datos Personales”. 2019.
- [5] C. Troncoso, “Privacy & Online Rights Knowledge Area”, 2019.
- [6] T. Ermakova, B. Fabian, y H. Krasnova, “Privacy Policies and Users ’ Trust : Does Readability Matter ?”, núm. Pollach 2007, pp. 1–12, 2014.
- [7] S. Wilson *et al.*, “The Creation and Analysis of a Website Privacy Policy Corpus”, 2016.
- [8] S. Wilson *et al.*, “Analyzing Privacy Policies at Scale: From Crowdsourcing to Automated Annotations”, *ACM Trans. Web*, vol. 13, núm. 1, 2018.
- [9] G. Contissa *et al.*, “Automated processing of privacy policies under the EU general data protection regulation”, *Front. Artif. Intell. Appl.*, vol. 313, núm. see 4, pp. 51–60, 2018.
- [10] H. Harkous, K. Fawaz, R. Leuret, F. Schaub, K. G. Shin, y K. Aberer, *Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning*, Vol 27. UNESIX, 2018.
- [11] P. Story *et al.*, “Natural Language Processing for Mobile App Privacy Compliance”.
- [12] T. Libert, “An Automated Approach to Auditing Disclosure of Third-Party Data Collection in Website Privacy Policies”, 2018.
- [13] Naciones Unidas, “Carta Internacional de los derechos del hombre Declaración universal de derechos del hombre”. 1948.

- [14] D. J. Solove, “A taxonomy of privacy”, *Calif. Law Rev.*, vol. 90, núm. 4, pp. 1087–1155, 2002.
- [15] M. Madejski y S. M. Bellovin, “The Failure of Online Social Network Privacy Settings”, *Methodology*, vol. 010, núm. CUCS-010-11, pp. 1–20, 2011.
- [16] G. Fawkes, “Report: Ecuadorian Breach Reveals Sensitive Personal Data”. [En línea]. Disponible en: <https://www.vpnmentor.com/blog/report-ecuador-leak/>. [Consultado: 22-oct-2019].
- [17] República del Ecuador, “Constitución del Ecuador”, *Const. del Ecuador*, p. 132, 2008.
- [18] Ecuador, “Ley de comercio electrónico, firmas y mensajes de datos”, pp. 1–17, 2002.
- [19] Ecuador, “Ley del Sistema Nacional de Registro de Datos Públicos”, *J. Chem. Inf. Model.*, vol. 53, núm. 9, pp. 1689–1699, 2018.
- [20] República del Ecuador, “Codigo Organico Integral Penal”, *Noticias*, p. 1, 2018.
- [21] INEC, “Encuesta Calidad De Vida 2015”, 2015.
- [22] K. Schwaber y J. Sutherland, “The Scrum Guide: The Definitive The Rules of the Game”, *Scrum.Org and ScrumInc*, núm. November, p. 19, 2017.
- [23] G. Chrome, “MVC Architecture”. [En línea]. Disponible en: https://developer.chrome.com/apps/app_frameworks. [Consultado: 08-jul-2020].
- [24] C. Bird y E. T. Barr, “To type or not to type: Quantifying Detectable bugs in JavaScript”, 2017.
- [25] Python, “What is Python? Executive Summary”. [En línea]. Disponible en: <https://www.python.org/doc/essays/blurb/>. [Consultado: 30-jul-2020].
- [26] “Flask | The Pallets Projects”. [En línea]. Disponible en: <https://palletsprojects.com/p/flask/>. [Consultado: 31-jul-2020].
- [27] M. Cohn, “Agile estimating and planning”, *HortScience*, vol. 47, núm. 12, 2005.
- [28] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, vol. 1, 2004.
- [29] L. Richardson, M. Amundsen, y S. Ruby, *RESTful Web APIs*, 2013.
- [30] Flask, “API Docs — Flask-RESTful 0.3.8 documentation”. [En línea]. Disponible en:

<https://flask-restful.readthedocs.io/en/latest/api.html>. [Consultado: 16-ago-2020].

- [31] RXJS, “RxJS - Observable”. [En línea]. Disponible en:
<https://rxjs.dev/api/index/class/Observable>. [Consultado: 31-ago-2020].
- [32] RXJS, “RxJS - BehaviorSubject”. [En línea]. Disponible en:
<https://rxjs.dev/api/index/class/BehaviorSubject>. [Consultado: 31-ago-2020].
- [33] K. Krippendorff, “Computing Krippendorff ’ s Alpha-Reliability”, *Dep. Pap.*, p. 12, 2011.
- [34] Angular, “Angular - Introduction to Angular concepts”. [En línea]. Disponible en:
<https://angular.io/guide/architecture>. [Consultado: 29-jul-2020].
- [35] Flask, “Application Factories — Flask Documentation (1.1.x)”. [En línea]. Disponible en: <https://flask.palletsprojects.com/en/1.1.x/patterns/appfactories/>. [Consultado: 06-ago-2020].
- [36] Material, “Introduction - Material Design”. [En línea]. Disponible en:
<https://material.io/design/introduction#principles>. [Consultado: 10-ago-2020].

6. ANEXOS

Esta sección presenta un resumen de los anexos que fueron necesarios para este proyecto.

Los anexos utilizados son los siguientes:

ANEXO A: Listado completo de las historias de usuario.

ANEXO B: Interfaces de usuario (mockups) completas.

ANEXO C: Explicación detallada de Angular.

ANEXO D: Arranque de aplicación *backend*

ANEXO E: Arranque de aplicación *frontend*

ANEXO F: Código implementado para notificación de inconsistencia en anotación.

ANEXO G: Notificación de sección inconsistente en consolidación.

ANEXO H: Código *frontend* y manual de usuario de la herramienta de anotación.

ANEXO I: Código *frontend* y manual de usuario de la herramienta de visualización.

ANEXO J: Código *backend* de la herramienta

ANEXO K: Criterios de aceptación y evidencias.

ANEXO L: Cálculo manual coeficiente inter-annotador.

ANEXO M: Encuesta de satisfacción herramienta de anotación y visualización.

ANEXO A

Listado completo de las historias de usuario.

En este se presentan todas las historias de usuario generadas para la implementación de las herramientas de anotación y visualización.

Tabla A.1 Historia de usuario HU-01

Nombre: Inicio y cierre de sesión	
Código: HU-01	Asignada: Félix Miño
Prioridad: Alta	Estimación: 3
Descripción: Como anotador o administrador de la herramienta requiero que la herramienta de anotación cuente con un mecanismo para el inicio y cierre de sesión de esta manera se evitará accesos no autorizados.	
Criterios de aceptación: Como anotador o administrador requiero estar en la capacidad de: <ul style="list-style-type: none">• Iniciar sesión al ingresar mi usuario (correo electrónico) y contraseña.• Cerrar sesión.	

Tabla A.2 Historia de usuario HU-02

Nombre: Administración de usuarios en la herramienta	
Código: HU-02	Asignada: Félix Miño
Prioridad: Media	Estimación: 5
Descripción: Como administrador requiero que la herramienta de permita la administración de usuarios de esta manera puedo tener control sobre los usuarios que están registrados en la herramienta. Así mismo, requiero administrar roles de usuario y módulos.	
Criterios de aceptación: Como administrador de la herramienta requiero: <ul style="list-style-type: none">• Administrar roles de usuario de usuario y módulos.• Crear un usuario. El usuario debe tener los campos: correo electrónico, rol de usuario y entrenamiento (valor booleano). No debe permitirse campos en blanco.• Elegir el rol de usuario (anotador o administrador) al momento de crear uno.• Elegir si un usuario estará en entrenamiento o no.• Al momento de crear un usuario, este nuevo usuario sea notificado a su correo electrónico de su rol de usuario en la herramienta y sus credenciales de acceso. La clave será autogenerada• Visualizar una lista de los usuarios presentes en la herramienta.• Editar los atributos de un usuario: correo electrónico, rol de usuario, si el usuario se encuentra en entrenamiento o no y si el usuario está activo o no en la herramienta.• Eliminar un usuario. En caso de que no sea posible eliminarlo, se requiere marcarlo como usuario inactivo, lo que restringirá el acceso a la herramienta.	

Tabla A.3 Historia de usuario HU-03

Nombre: Administración de tratamientos de datos	
Código: HU-03	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
<p>Descripción: Como administrador requiero que la herramienta permita la administración de tratamientos de datos de esta manera puedo tener control sobre los tratamientos de datos con los que se realizarán las anotaciones. Así mismo, requiero administrar los colores que serán asignados a los tratamientos.</p>	
<p>Criterios de aceptación: Como administrador de la herramienta requiero:</p> <ul style="list-style-type: none"> • Administrar los colores que serán asignados a los tratamientos de datos. • Crear un tratamiento de datos. El tratamiento de datos debe tener: descripción y color. No deben permitirse campos en blanco. • Visualizar los tratamientos de datos existentes. • Editar un tratamiento de datos: descripción y color. • Eliminar un tratamiento de datos. Manteniendo la integridad referencial. • Filtrar un tratamiento de datos en base a la descripción. 	

Tabla A.4 Historia de usuario HU-04

Nombre: Administración de atributos	
Código: HU-04	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
<p>Descripción: Como administrador requiero que la herramienta permita la administración de atributos de esta manera puedo tener control sobre los atributos con los que se realizarán las anotaciones.</p>	
<p>Criterios de aceptación: Como administrador de la herramienta requiero:</p> <ul style="list-style-type: none"> • Crear un atributo y relacionarlo a un tratamiento de datos. Un atributo tendrá: descripción. • Visualizar los atributos pertenecientes a un tratamiento de datos. • Editar un atributo: descripción. • Eliminar un atributo. Manteniendo la integridad referencial. 	

Tabla A.5 Historia de usuario HU-05

Nombre: Administración de valores	
Código: HU-05	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
Descripción: Como administrador requiero que la herramienta permita la administración de valores de un atributo de esta manera puedo tener control sobre los valores con los que se realizarán las anotaciones.	
Criterios de aceptación: Como administrador de la herramienta requiero: <ul style="list-style-type: none"> • Crear un valor y relacionarlo a un atributo. Un valor tendrá: descripción. • Visualizar los valores pertenecientes a un atributo de un tratamiento de datos. • Editar un valor: descripción. • Eliminar un valor. Manteniendo la integridad referencial. 	

Tabla A.6 Historia de usuario HU-06

Nombre: Administración de políticas de privacidad	
Código: HU-06	Asignada: Félix Miño
Prioridad: Media	Estimación: 5
Descripción: Como administrador requiero que la herramienta permita la administración de políticas de privacidad de esta manera puedo tener control sobre las políticas a ser anotadas y los usuarios asignados a anotar y consolidar.	
Criterios de aceptación: Como administrador de la herramienta requiero: <ul style="list-style-type: none"> • Crear una política de privacidad. La política tendrá: nombre, URL de donde se obtuvo la política, fecha en la se obtuvo la política y un archivo en formato "txt". • Dar formato al texto presente en el archivo "txt". El carácter "+" precede a un título y dos saltos de línea representan una nueva sección. • Previsualizar la política de privacidad que será guardada. • Visualizar las políticas que fueron creadas. • Filtrar una política de privacidad por nombre o URL. • Asignar una política a: 1) mínimo dos anotadores y 2) un administrador quien será el que realice la consolidación. • Editar una política de privacidad: nombre, URL y fecha. • Eliminar una política de privacidad. Manteniendo la integridad referencial. 	

Tabla A.7 Historia de usuario HU-07

Nombre: Visualización de políticas de privacidad por anotar	
Código: HU-07	Asignada: Félix Miño
Prioridad: Media	Estimación: 2
Descripción: Como anotador o administrador requiero que la herramienta me permita observar las políticas de privacidad pendiente de anotar de esta manera puedo tener seguimiento de las políticas de privacidad que me fueron asignadas.	

Criterios de aceptación:

Como anotador o administrador de la herramienta requiero:

- Visualizar las políticas de privacidad que me fueron asignadas para anotar y aún no fueron terminadas.
- Junto a cada política de privacidad visualizar el porcentaje de progreso.

Tabla A.8 Historia de usuario HU-08

Nombre: Anotación de una política de privacidad	
Código: HU-08	Asignada: Félix Miño
Prioridad: Alta	Estimación: 8
Descripción: Como anotador o administrador requiero que la herramienta me permita realizar anotaciones de los tratamientos de datos que presentan las políticas de privacidad de esta manera puedo codificar una política y almacenar estos resultados de forma permanente en una base de datos.	
Criterios de aceptación: Como anotador o administrador de la herramienta requiero: <ul style="list-style-type: none"> • Al dar <i>click</i> en una política de privacidad en la sección visualización ser redirigido a la página para la anotación de la misma. • Tener una lista de los tratamientos de datos, sus atributos y sus valores. Solo se presentarán los tratamientos que incluyan al menos un atributo y un valor. • Observar la política separada por las secciones que fueron predefinidas. • Guardar una anotación. Una anotación consiste en la combinación de: uno o más valores, una sección de texto, un valor que identifica si la anotación ejecuta o no las acciones presentadas en la política y un comentario (opcional), sin embargo, el comentario no será cubierto por esta historia de usuario. • Cuando se haya llegado a la última sección de la política y se de <i>click</i> en “siguiente” la herramienta debe notificar que se llegó al final del documento y requiero poder elegir si terminar o no la codificación. 	

Tabla A.9 Historia de usuario HU-09

Nombre: Visualización, edición y eliminación de anotaciones	
Código: HU-09	Asignada: Félix Miño
Prioridad: Media	Estimación: 5
Descripción: Como anotador o administrador requiero que la herramienta me permita visualizar, editar y eliminar anotaciones previamente realizadas de esta manera tengo mayor control sobre la codificación de una política de privacidad.	
Criterios de aceptación: Como anotador o administrador de la herramienta requiero: <ul style="list-style-type: none"> • Tener un conteo del número de anotaciones totales que realicé sobre una sección de una política de privacidad. • Visualizar las anotaciones que realicé sobre una sección. • Editar una anotación previamente realizada. Se podrá editar: la acción (ejecuta o no ejecuta) y el comentario de una anotación. Los valores y el texto están fuera del alcance. • Eliminar una anotación que realicé. 	

Tabla A.10 Historia de usuario HU-10

Nombre: Notificación de incosistencia en anotación	
Código: HU-10	Asignada: Félix Miño
Prioridad: Media	Estimación: 5
<p>Descripción: Como anotador o administrador (en entrenamiento) requiero que la herramienta me notifique en caso de existir inconsistencia al momento de realizar una anotación de esta manera las anotaciones realizadas por los usuarios a lo largo de una política serán consistentes.</p>	
<p>Criterios de aceptación: Como anotador o administrador (en entrenamiento) de la herramienta requiero:</p> <ul style="list-style-type: none"> • Ser notificado en caso de que uno o más de los valores que estoy usando para realizar la anotación no se encuentren anotados por otro usuario en esa sección de la política. Esto para cada acción (ejecuta o no ejecuta). • Tener una lista de posibles tratamientos (sugerencias) para realizar la anotación. Esto se hará en base a valores previamente anotados por otros usuarios. En caso de esa sección haya sido dejada sin anotaciones de manera intencional, en lugar de una lista de sugerencia de valores, requiero tener una sugerencia de dejar esta sección sin anotar. 	

Tabla A.11 Historia de usuario HU-11

Nombre: Visualización de políticas de privacidad para consolidación	
Código: HU-11	Asignada: Félix Miño
Prioridad: Media	Estimación: 1
<p>Descripción: Como administrador requiero que la herramienta me permita observar las políticas de privacidad que me fueron asignadas para consolidar de esta manera puedo hacer seguimiento de estas.</p>	
<p>Criterios de aceptación: Como administrador de la herramienta requiero:</p> <ul style="list-style-type: none"> • Visualizar en una lista las políticas de privacidad que me fueron asignadas para consolidar y aún no fueron terminadas. • Visualizar una política de privacidad solamente cuando esta fue terminada de anotar por todos los usuarios asignados para esa tarea. • Junto a cada política de privacidad visualizar el porcentaje de progreso. 	

Tabla A.12 Historia de usuario HU-12

Nombre: Cálculo de coeficiente inter-anotador	
Código: HU-12	Asignada: Félix Miño
Prioridad: Media	Estimación: 8
<p>Descripción: Como administrador requiero que la herramienta calcule un coeficiente inter-anotador de esta manera puedo tener una referencia de que tan confiables son las anotaciones realizadas por los usuarios en una política de privacidad.</p>	
<p>Criterios de aceptación: Como administrador de la herramienta requiero:</p> <ul style="list-style-type: none"> • En la página de visualización de políticas de privacidad por consolidar, tener una opción para ver el coeficiente inter-anotador y el número total de anotaciones que cada usuario realizó sobre la política. El coeficiente será calculado con el índice Alpha de Krippendorff. 	

Tabla A.13 Historia de usuario HU-13

Nombre: Consolidación de una política de privacidad	
Código: HU-13	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
<p>Descripción: Como administrador requiero que la herramienta me permita realizar anotaciones, de consolidación, de los tratamientos de datos que presentan las políticas de privacidad de esta manera puedo codificar una política y almacenar estos resultados de forma permanente en una base de datos.</p>	
<p>Criterios de aceptación: Como administrador de la herramienta requiero:</p> <ul style="list-style-type: none"> • Al dar <i>click</i> en una política de privacidad en la sección de consolidación ser redirigido a la página para la consolidación de la misma. • Tener una lista de los tratamientos de datos, sus atributos y valores. Solo se presentarán los tratamientos que incluyan al menos un atributo y un valor. • Seleccionar uno o múltiples valores para realizar una anotación. • Observar la política separada por las secciones que fueron predefinidas. • Seleccionar una sección de texto de una de las secciones de la política de privacidad para realizar una anotación. • Seleccionar si la anotación ejecuta o no los tratamientos de datos expresados en la misma. • Guardar una anotación. Una anotación consiste en la combinación de: uno o más valores, una sección de texto, un valor que identifica si la anotación ejecuta o no las acciones presentadas en la política y un comentario (opcional), sin embargo, el comentario no será cubierto por esta historia de usuario. • Cuando se haya llegado a la última sección de la política y se de <i>click</i> en “siguiente” la herramienta debe notificar que se llegó al final del documento y requiero poder elegir si terminar o no la codificación. 	

Tabla A.14 Historia de usuario HU-14

Nombre: Visualización, edición y eliminación de anotaciones en consolidación	
Código: HU-14	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
Descripción: Como administrador requiero que la herramienta me permita visualizar, editar y eliminar anotaciones de consolidación previamente realizadas de esta manera tengo mayor control sobre la codificación de una política de privacidad.	
Criterios de aceptación: Como administrador de la herramienta requiero: <ul style="list-style-type: none"> • Tener un conteo del número total de anotaciones que realicé sobre una sección de una política de privacidad en la etapa de consolidación. • Visualizar las anotaciones que realicé sobre una sección en consolidación. • Editar una anotación previamente realizada en consolidación. Se podrá editar: la acción (ejecuta o no ejecuta) y el comentario de una anotación. Los valores y el texto están fuera del alcance. • Eliminar una anotación que realicé previamente en consolidación. 	

Tabla A.15 Historia de usuario HU-15

Nombre: Visualización y copia de anotaciones de la etapa de anotación en consolidación	
Código: HU-15	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
Descripción: Como administrador requiero que la herramienta me permita visualizar y copiar las anotaciones realizadas por los usuarios en la etapa de anotación de esta manera puedo ahorrar tiempo al momento de realizar las anotaciones de consolidación.	
Criterios de aceptación: Como administrador de la herramienta requiero: <ul style="list-style-type: none"> • Visualizar cada usuario que realizó la etapa de anotación con sus respectivas anotaciones, en cada sección de la política de privacidad. • Copiar una o más de las anotaciones realizadas por los usuarios en la etapa de anotación y que cuente como una anotación de consolidación. 	

Tabla A.16 Historia de usuario HU-16

Nombre: Notificación de seccion incositente en consolidación	
Código: HU-16	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
Descripción: Como administrador requiero que la herramienta me permita ver que secciones de una política de privacidad son inconsistentes y cuáles no, de esta manera al momento de consolidar tengo una visión más clara de que secciones necesitan mayor atención.	
Criterios de aceptación: Como administrador de la herramienta requiero:	

- Visualizar cada sección de una política de privacidad con una notificación de consistencia o inconsistencia. Se considera que una sección es consistente cuando todas las anotaciones de todos los usuarios son iguales, en cuanto a tratamientos de datos y acción (ejecuta o no ejecuta), no se tomará en cuanto el texto y comentario. También se considera una sección consistente si no presenta ninguna anotación.

Tabla A.17 Historia de usuario HU-17

Nombre: Pagina de bienvenida en herramienta de visualización	
Código: HU-17	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
Descripción: Como usuario de la herramienta de visualización requiero tener una página de bienvenida de esta manera puedo tener una visión general del funcionamiento de la herramienta.	
Criterios de aceptación: Como usuario de la herramienta de visualización requiero: <ul style="list-style-type: none"> • Al ingresar a la herramienta de visualización tener una página de bienvenida que me da una visión general de este proyecto. 	

Tabla A.18 Historia de usuario HU-18

Nombre: Lista de políticas de privacidad para visualizar	
Código: HU-18	Asignada: Félix Miño
Prioridad: Media	Estimación: 3
Descripción: Como usuario de la herramienta de visualización requiero tener una lista de las políticas de privacidad que se encuentran listas para ser visualizadas, es decir que concluyeron con el proceso de anotación, de este modo tengo esta información de manera estructurada.	
Criterios de aceptación: Como usuario de la herramienta de visualización requiero: <ul style="list-style-type: none"> • Ver las políticas que concluyeron el proceso de anotación en forma de lista. Una fila de una política de privacidad incluirá: índice, nombre, fecha en que se anotó y el número total de anotaciones. • Filtrar una política de privacidad por su nombre. 	

Tabla A.19 Historia de usuario HU-19

Nombre: Visualización de una política de privacidad anotada	
Código: HU-19	Asignada: Félix Miño
Prioridad: Alta	Estimación: 8
<p>Descripción: Como usuario de la herramienta de visualización requiero visualizar una política de privacidad que pasó por el proceso de anotación de esta manera puedo observar, de una manera sencilla, los tratamientos de datos personales que el sitio dueño de la política de privacidad lleva a cabo sobre mis datos.</p>	
<p>Criterios de aceptación: Como usuario de la herramienta de visualización requiero:</p> <ul style="list-style-type: none"> • Al dar <i>click</i> en una política de la lista “Políticas de privacidad anotadas” ser redirigido a la página de visualización y observar resaltados los tratamientos de datos que fueron anotados sobre ésta. • Observar una anotación resaltada por el color del primer tratamiento de datos (en caso de que sea una anotación multivalor). • Al poner el cursor sobre una anotación me muestre un cuadro de texto que contenga: el número de tratamientos que conforman la anotación, la acción (ejecuta o no ejecuta), una lista detalla (tratamiento, atributo y valor) para cada valor de la anotación y el comentario con el que fue anotado (en caso de existir). • Observar una lista de todos los tratamientos con los que se pudo anotar la política, es decir un tratamiento que al menos tiene un atributo y un valor. • Observar junto a cada tratamiento el número de anotaciones que lo contienen y una barra del porcentaje que representa el uso de este tratamiento sobre el total de tratamientos presentes en la política. • Al dar <i>click</i> en un tratamiento de la lista expandir una lista de los atributos que contiene este y al dar <i>click</i> nuevamente colapsar los atributos. 	

Tabla A.20 Historia de usuario HU-20

Nombre: Filtrado de anotaciones en visualización	
Código: HU-20	Asignada: Félix Miño
Prioridad: Media	Estimación: 5
<p>Descripción: Como usuario de la herramienta de visualización requiero tener la opción para filtrar las anotaciones presentes en una política de privacidad por el tratamiento o atributo de esta manera puedo tener anotaciones más específicas e información más refinada.</p>	
<p>Criterios de aceptación: Como usuario de la herramienta de visualización requiero:</p> <ul style="list-style-type: none"> • Al dar <i>click</i> sobre uno de los tratamientos listados en la página de visualización, la política de privacidad se filtre y muestre solamente las anotaciones que incluyan este tratamiento. Al dar <i>click</i> nuevamente sobre el tratamiento mostrar la política de privacidad sin filtros y colapsar los atributos. • Al dar <i>click</i> en un atributo de un tratamiento, filtrar la política de privacidad con las anotaciones que contengan ese atributo. Se mostrará un botón que permitirá limpiar el filtrado y mostrará la política de privacidad original. • El filtrado a nivel multitratamiento o multiatributo está fuera del alcance de este proyecto. 	

ANEXO B

Los mockups de las interfaces visuales se encuentran en el anexo digital. El archivo `herramienta_annotacion.xd` corresponde a la herramienta de anotación , mientras que el archivo `herramienta_visualizacion.xd` corresponde a la herramienta de visualización.

También en el anexo digital se presenta un archivo `.pdf` generado desde Adobe XD, para cada una de las herramientas.

ANEXO C

Explicación detallada de Angular

Angular es un *framework* de *frontend* para construir aplicaciones de una sola página (SPA, *Single Page Application*) haciendo uso de *HTML*, *CSS* y *TypeScript*. La arquitectura de una aplicación de Angular se basa en tres conceptos fundamentales: módulos, componentes y servicios. Los módulos, componentes y servicios son clases que usan el patrón de diseño Decorador. Un decorador en Angular permite marcar cada una de estas clases con su tipo correspondiente y provee metadata para que Angular pueda interpretar y hacer uso de estas.

La unidad básica para construir un sitio con Angular son los módulos, llamados *NgModules*. Proporcionan a los componentes un contexto para que puedan ser compilados. Los módulos reúnen código que está relacionado y lo unen en conjuntos funcionales, una aplicación construida con Angular se define mediante un conjunto de módulos. Una aplicación tiene al menos un módulo raíz, llamado *AppModule* que permite arrancar la aplicación, y por lo general tiene más módulos que lo caracterizan y se crean de acuerdo con las necesidades del proyecto. El decorador usado por un módulo es `@NgModule`.

Así mismo, una aplicación Angular está construida por un conjunto de componentes. Al igual que los módulos, una aplicación tiene al menos un componente llamado componente raíz y conecta la jerarquía de componentes al Modelo de Objetos del Documento (DOM, *Document Object Model*). El decorador usado por un componente es `@Component`. Los componentes de Angular ejecutan básicamente dos acciones:

- Definen vistas al unirse con plantillas. Las plantillas que son conjuntos de elementos visuales (HTML) que Angular puede acceder y modificar mediante directivas de acuerdo con la lógica definida en el componente. Las directivas en una plantilla proveen la lógica del programa.
- Usan servicios los cuales proveen funcionalidad que no está directamente relacionada con la vista, por ejemplo, una petición al servidor.

Por último, los servicios son una amplia categoría que abarca cualquier valor, función o característica que una aplicación necesita. Para cualquier lógica o datos que no están necesariamente asociados con una vista y que será usada a través de distintos componentes se puede definir una clase servicio. Una de las ventajas de los servicios es

que pueden inyectarse en los componentes como dependencias, de esta manera un mismo servicio puede ser usado en varios componentes haciendo que su código sea modular, *reutilizable* y eficiente. El decorador usado por un servicio es `@Injectable`.

Otro concepto importante en Angular es el *data binding*. Sin el uso de un *framework* de *frontend* como Angular, la inserción de datos en las vistas *HTML* y la conversión de las interacciones de los usuarios en actualizaciones de los valores que se muestran en las vistas tendrían que ser hechas de manera manual. El escribir esta lógica de *pull and push* sería tedioso, propenso a errores y resultaría en código complicado de entender y mantener [34]. Angular soporta *two-way data binding* un mecanismo usado para coordinar las partes de una vista con las partes de un componente. La Figura C.1 muestra las cuatro formas de *data binding* que existe en Angular.

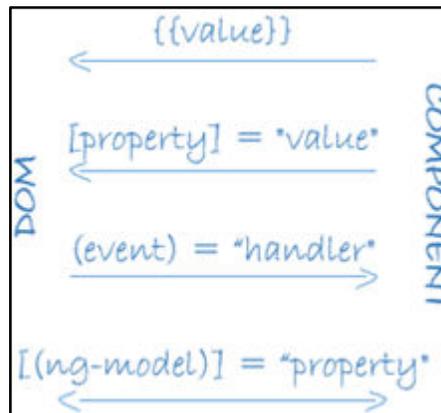


Figura C.1 *Data binding* en Angular [34]

1. `{{value}}`: conocido como *interpolation*, permite mostrar un valor del componente en un elemento de la vista.
2. `[property]="value"`: es llamado *property binding* y es usado para enlazar una propiedad con un valor.
3. `(event)="handler"`: conocido como *event binding*, permite asociar una función específica a una interacción del usuario con la vista, por ejemplo: dar *click* en un botón.
4. `[(ng-model)]="property"`: este tipo de declaración es conocida como *two-way data binding* y combina *property* y *event binding* en una sola declaración.

El *data binding* juega un papel fundamental en la comunicación de la plantilla *HTML* con su componente y también entre un componente padre y su hijo. La Figura C.2 ilustra estas dos comunicaciones.

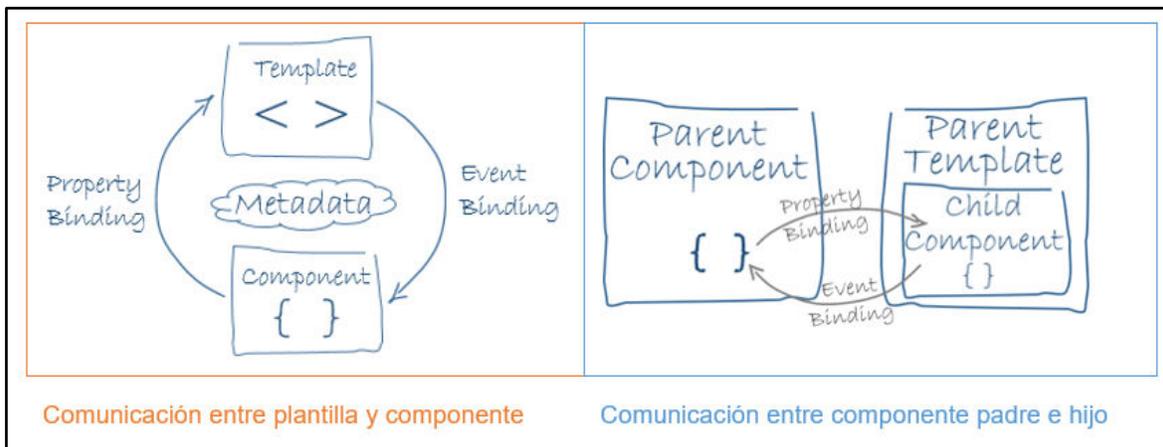


Figura C.2 Tipos de comunicaciones con *data binding* [34]

Finalmente, todo lo expuesto en esta sección se encuentra resumido en la Figura C.3.

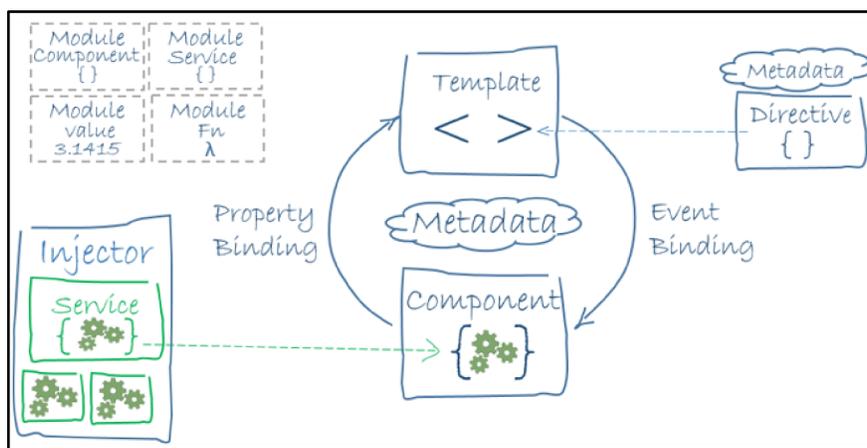


Figura C.3 Vista general de Angular [34]

De la figura anterior se puede resaltar lo siguiente:

- Un componente en conjunto con una plantilla define una vista de Angular.
 - El decorador en una clase componente añade metadata necesaria para que Angular lo interprete.
 - Las directivas y el *binding markup* en la plantilla de un componente permiten modificar las vistas en base a la lógica y los datos que tiene la aplicación.

La inyección de dependencias provee un servicio a un componente.

ANEXO D

Arranque de la aplicación *Backend*

El Código D.1 corresponde al archivo `manage.py` el cual ejecuta la aplicación de *backend*. Está distribuido de la siguiente manera:

- En las líneas 1-5 se realizan las importaciones tanto de las librerías como del archivo `main` en la línea 3.
- En la línea 8 se declara una variable que contiene el ambiente en el que se está trabajando, puede ser “*dev*” para desarrollo o “*prod*” para producción.
- En la línea 10 se crea la aplicación llamando a la función `crear_aplicacion` del archivo `main`. Esta función será presentada en el código 2.4.
- La línea 12 permite registrar los diferentes *endpoints*⁴⁵ de la aplicación, es decir los puntos de comunicación de las APIs RESTFul.
- La línea 14 permite enlazar la aplicación con un objeto SQLAlchemy. Esto es necesario en caso de que existan dos o más aplicaciones en el servidor, sin embargo, se añadió este código como buena práctica.
- En la línea 16 se instancia la clase `Manager` que permite correr ciertos comandos desde la terminal. Por ejemplo: `Python manage.py run` que permite correr la aplicación `flask`.
- De la línea 18-20 corresponde a las migraciones de datos que hará la aplicación. La línea 20 realiza una instancia de la `Migrate`, que contiene todas las funcionalidades para las migraciones.
- En la línea 23 se declara los sitios que pueden realizar peticiones al servidor, en este caso para fines de desarrollo se habilitó todos los orígenes. Esta característica es conocida como CORS ⁴⁶.
- Finalmente, de la línea 26 a la 33 se declaran las dos funciones que permiten que la aplicación sea ejecutable desde la línea de comandos. Para iniciar la aplicación es necesario ejecutar el comando: `python manage.py run`.

⁴⁵ **Endpoint:** es una interfaz expuesta para la comunicación. Para el caso de una API RESTFUL un endpoint se identifica de manera por la ruta del recurso.

⁴⁶ **CORS:** intercambio de recursos de origen cruzado, es un mecanismo que permite solicitar recurso desde un dominio diferente al que se encuentra el recurso.

```

1  from flask_migrate import Migrate, MigrateCommand
2  from flask_script import Manager
3  from app.main import crear_aplicacion, db
4  from app import blueprint
5  from flask_cors import CORS
6
7
8  ENTORNO = 'dev'
9
10 app = crear_aplicacion(ENTORNO)
11
12 app.register_blueprint(blueprint)
13
14 app.app_context().push()
15
16 manager = Manager(app)
17
18 manager.add_command('db', MigrateCommand)
19
20 migrate = Migrate(app, db, compare_type=True)
21
22
23 CORS(app, resources='/*')
24
25
26 @manager.command
27 def run():
28     app.run()
29
30
31 if __name__ == '__main__':
32     manager.run()
33

```

Código D.1 Arranque de la aplicación 1 de 3

El Código D.2 corresponde al archivo `app/__init__.py`. En este archivo se crea el objeto Flask haciendo uso del patrón *factory* definido en la documentación oficial de flask [35]. De esta manera se puede tener varias instancias de flask, es decir se puede hacer uso de una instancia específica para cada ambiente en el que se desea la aplicación, en este caso se hará uso de los ambientes de desarrollo y producción. Cada instancia puede tener una configuración para la base de datos y características propias, por ejemplo, permitir la depuración de errores.

```

1  from flask import Flask
2  from flask_sqlalchemy import SQLAlchemy
3  from flask_bcrypt import Bcrypt
4  from .config import config_by_name
5
6  db = SQLAlchemy()
7  flask_bcrypt = Bcrypt()
8
9
10 def crear_aplicacion(config_name):
11     app = Flask(__name__)
12     app.config.from_object(config_by_name[config_name])
13     db.init_app(app)
14     flask_bcrypt.init_app(app)
15
16     return app
17

```

Código D.2 Arranque de la aplicación 2 de 3

En el Código D.3 se expone los diferentes ambientes en los que puede arrancar la aplicación. Se excluyó algunas líneas de este archivo ya que incluyen “secretos”⁴⁷ del código.

```

13 class DevelopmentConfig(Config):
14     SQLALCHEMY_DATABASE_URI = mysql_local_base
15     DEBUG = True
16     SQLALCHEMY_TRACK_MODIFICATIONS = False
17
18
19 class ProductionConfig(Config):
20     DEBUG = False
21     SQLALCHEMY_DATABASE_URI = mysql_aws
22     SQLALCHEMY_TRACK_MODIFICATIONS = True
23
24
25     config_by_name = dict(
26         dev=DevelopmentConfig,
27         prod=ProductionConfig
28     )
29
30     key = Config.SECRET_KEY
31

```

Código D.3 Arranque de la aplicación 3/3

El código presentado en esta sección es suficiente para arrancar una aplicación Flask.

⁴⁷ Se considera secretos del código cualquier información que permite acceso a servidores, bases de datos o en general a código que debe estar limitado dado su nivel de confidencialidad.

ANEXO E

Arranque de la aplicación *frontend*

Angular permite crear una aplicación desde la línea de comandos. Al correr el comando `ng new nombre-proyecto` se creará un proyecto con los directorios y archivos necesarios para arrancar una aplicación básica. En la Figura E.1 el directorio `src` contiene los directorios y archivos que forman la aplicación. En este directorio se incluirá los componentes necesarios para crear la aplicación deseada.

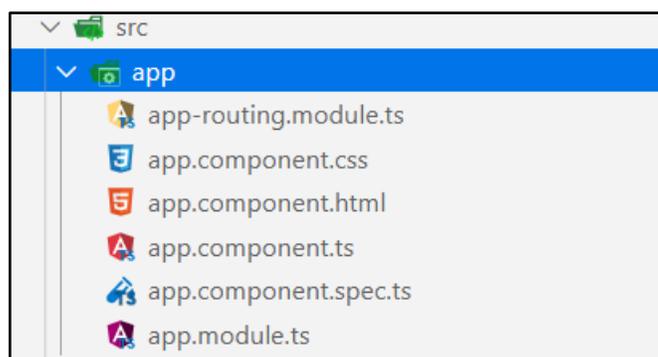


Figura E.1 Estructura básica de aplicación Angular

Una vez generada la aplicación se puede iniciar mediante el comando `ng server`. Angular por defecto usa el puerto 4200. En la Figura E.2 se presenta la pantalla de la aplicación base generada por Angular.

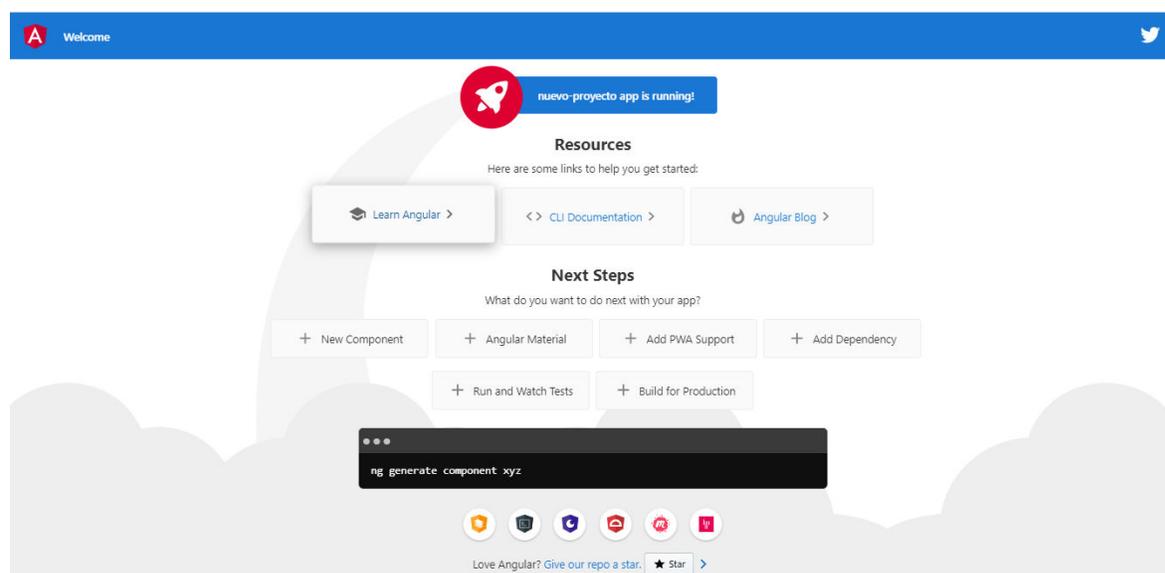


Figura E.2 Pantalla de inicio de aplicación Angular básica

En este proyecto se hará uso de la librería Angular Material que es una colección de componentes web basados en Material Design, una guía de estilo creada por Google para

construir aplicaciones móviles y web [36]. Para instalar Angular Material es necesario ejecutar el comando `ng add @angular/material`. Algunas de las acciones que este comando ejecuta son:

- Añade las dependencias necesarias en el archivo `package.json`.
- Añade la fuente `Roboto` en el archivo `index.html`.
- Añade la fuente de íconos de Material Design en el archivo `index.html`.
- Añade algunos estilos de CSS a nivel global:
 - Remueve los márgenes de elemento `body`.
 - Establece la propiedad `height:100%` en los elementos `html` y `body`.
 - Establece la fuente `Roboto` como predeterminada.

Finalmente, en el directorio `app` se añade el archivo `material.ts` que contiene los componentes que ofrece material y los exporta como módulo. Este módulo es incluido en el archivo `app.module.ts` para que estos elementos puedan ser accedidos de manera global en el proyecto y que cualquier otro componente pueda hacer uso de los mismos.

ANEXO F

Código implementado para la notificación de inconsistencias en anotación de usuarios en entrenamiento.

En este anexo se presenta el código de *backend* y *frontend* implementado para la notificación de inconsistencias en la etapa de anotación para usuarios (anotadores o administradores) que se encuentren en entrenamiento.

El **Código F.1** se ejecuta en el *frontend* de la aplicación. Primero se consulta desde el *backend* si el usuario que está realizando la anotación se encuentra en entrenamiento, en caso afirmativo se consulta si existe inconsistencia en las anotaciones que desea guardar. Si el usuario no se encuentra en entrenamiento se guardan las anotaciones directamente.

```
103     this._usuarioService.obtenerUsuario(this.usuario.id).subscribe(  
104     usuario => {  
105         if (usuario.entrenamiento) {  
106             let anotacionNotificacion = new AnotacionNotificacion(this.usuario.id, this.parrafoId, !this.permite, this.valores)  
107             this._anotacionService.notificacionAnotacion(anotacionNotificacion).subscribe(  
108             (notificacion: AnotacionNotificacionConsultar) => {  
109                 if (notificacion.inconsistencia) {  
110                     let dialogoNotificacion = this._dialogo.open(NotificacionInconsistenciaComponent, {  
111                         width: '50%',  
112                         height: 'fit-content',  
113                         data: {  
114                             anotacion: anotacion,  
115                             notificacion: notificacion  
116                         }  
117                     })  
118  
119                     dialogoNotificacion.afterClosed().subscribe(  
120                     guardado => {  
121                         if (guardado) {  
122                             this.parrafoCambiado()  
123                             this._seleccionarTextoService.colocarTexto("")  
124                             this._seleccionarTextoService.colocarTextoHtml("")  
125                             this._seleccionarTextoService.consultarTotalAnotacionesAnotadorParrafoServicio(this.parrafoId, this.usuario.id)  
126                         } else {  
127                             this._seleccionarTextoService.colocarTexto(this.texto)  
128                             this._seleccionarTextoService.colocarTextoHtml(this.textoHtml)  
129                             this.lista.clear()  
130                             this.listaValores = []  
131                         }  
132                     }  
133                     )  
134  
135                     } else {  
136                         this.enviarAnotacion(anotacion)  
137                     }  
138                 }  
139             )  
140         } else {  
141             this.enviarAnotacion(anotacion)  
142         }  
143     },  
144     () => this.notificacion("ERROR creando anotacion!", "fracaso-snackbar")  
145 )  
146 };  
147 }
```

Código F.1 Notificación de inconsistencia en anotación, *frontend*

El Código F.2, Código F.3 y Código F.4 pertenecen al *backend* de la aplicación y solo se ejecutan si el usuario se encuentra en entrenamiento. En el Código F.2 se consulta todas las anotaciones que existen sobre la sección en la cual se desea guardar las anotaciones. En caso de que no existan anotaciones se ejecuta el Código F.3 y en caso de existir anotaciones se ejecuta el Código F.4. Cada sección se encuentra comentada para un mejor entendimiento.

```
100 def consultar_inconsistencia_notificacion(data):
101     #Se consulta todas las anotaciones reaizadas sobre esta sección
102     anotaciones_usuarios = (db.session.query(Anotacion, AnotacionValorRelacion, Valor, Atributo, Tratamiento)
103         .outerjoin(AnotacionValorRelacion, Anotacion.id == AnotacionValorRelacion.anotacion_id)
104         .outerjoin(Valor, AnotacionValorRelacion.valor_id == Valor.id)
105         .outerjoin(Atributo, Valor.atributo_id == Atributo.id)
106         .outerjoin(Tratamiento, Atributo.tratamiento_id == Tratamiento.id)
107         .filter(Anotacion.parrafo_id == data['parrafo_id'],
108             Anotacion.ejecuta == data['ejecuta'],
109             Anotacion.usuario_id != data['usuario_id'])
110         .order_by(Tratamiento.id).all())
```

Código F.2 Notificación de inconsistencia en anotación, *backend* 1 de 3

```

112 #En caso de que no existan valores sobre esta sección se realiza lo siguiente:
113 # Si alguno de los anotadores tiene anotaciones despues de la sección consultada
114 # o si alguno de los anotadores ha terminado la política de privacidad
115 # se sugiere que no se realicen anotaciones sobre esta sección
116 if not anotaciones_usuarios:
117     parrafo_secuencia = (db.session.query(Parrafo)
118         .filter(Parrafo.id == data['parrafo_id']).first())
119
120     usuarios_anotadores = (db.session.query(PoliticaUsuarioRelacion,)
121         .outerjoin(Politica, PoliticaUsuarioRelacion.politica_id == Politica.id)
122         .outerjoin(Parrafo, Politica.id == Parrafo.politica_id)
123         .filter(PoliticaUsuarioRelacion.consolidar == False,
124             Parrafo.id == data['parrafo_id'],
125             PoliticaUsuarioRelacion.usuario_id != data['usuario_id']).all())
126
127     usuarios_ultimo_parrafo = []
128
129     for politica_usuario in usuarios_anotadores:
130         usuarios_ultimo_parrafo.append(
131             consultar_ultima_anotacion_usuario_politica(politica_usuario.politica_id,
132                 politica_usuario.usuario_id,
133                 False))
134
135     i = 0
136     valores_no_consistentes = []
137     for valor in data['valores']:
138         valor = valor_notificacion(valor['valor_id'])
139         valores_no_consistentes.insert(i, valor[0])
140         valores_no_consistentes[i].valor_id = valor[0].id
141         valores_no_consistentes[i].valor_descripcion = valor[0].descripcion
142         valores_no_consistentes[i].atributo_descripcion = valor[1].descripcion
143         valores_no_consistentes[i].tratamiento_descripcion = valor[2].descripcion
144         valores_no_consistentes[i].color_primario = valor[2].color_tratamiento.codigo
145         i += 1
146
147
148     notificacion = AnotacionNotificacionConsultar(any(x > parrafo_secuencia.secuencia
149         for x in usuarios_ultimo_parrafo) or
150         any(x.finalizado == True for x in usuarios_anotadores),
151         valores_no_consistentes,
152         [])
153
154     return marshal(notificacion, AnotacionDto.anotacionNotificacionConsultar), 201

```

Código F.3 Notificación de inconsistencia en anotación, *backend* 2 de 3

```

156 #En caso de que si existan anotaciones sobre una sección se realiza lo siguientes
157 #se verifica si todos los valores enviados desde el frontend existen en las anotaciones
158 #antes consultadas, si todos existen la variable inconsistencia = False
159 #si alguno de los valores no existe se sugiere que no se anote bajo ese valor
160 #y se provee una lista que contiene todos los valores anotados sobre la sección
161 else:
162     consistencia_valores = []
163     valores_no_consistentes = []
164     valores_sugeridos = []
165     i = 0
166     for valor in data['valores']:
167         if any(x[1].valor_id == valor['valor_id'] for x in anotaciones_usuarios):
168             consistencia_valores.append(True)
169         else:
170             consistencia_valores.append(False)
171             valor = valor_notificacion(valor['valor_id'])
172             valores_no_consistentes.insert(i, valor[0])
173             valores_no_consistentes[i].valor_id = valor[0].id
174             valores_no_consistentes[i].valor_descripcion = valor[0].descripcion
175             valores_no_consistentes[i].atributo_descripcion = valor[1].descripcion
176             valores_no_consistentes[i].tratamiento_descripcion = valor[2].descripcion
177             valores_no_consistentes[i].color_primario = valor[2].color_tratamiento.codigo
178             i += 1
179
180     inconsistencia = not all(x for x in consistencia_valores)
181
182     if inconsistencia:
183         i = 0
184         for valor in anotaciones_usuarios:...
185
186     valores_sugeridos_unicos = lista_unica(valores_sugeridos)
187
188     notificacion = AnotacionNotificacionConsultar(inconsistencia,
189                                                    valores_no_consistentes,
190                                                    valores_sugeridos_unicos)
191
192     return marshal(notificacion, AnotacionDto.anotacionNotificacionConsultar), 201

```

Código F.4 Notificación de inconsistencia en anotación, *backend* 3 de 3

ANEXO G

Notificación de sección inconsistente en consolidación.

Esta funcionalidad permite tener una advertencia visual sobre cada sección de una política de privacidad acerca de la coincidencia o no de las anotaciones realizadas en la etapa de anotación por los distintos anotadores, es decir, si los anotadores realizaron las mismas anotaciones sobre una sección o no. Así mismo, se considera una coincidencia sobre una sección si no existe ninguna anotación.

En cuanto a la notificación existen tres posibles escenarios, para los ejemplos a continuación se considera que la política de privacidad fue asignada a dos anotadores:

1. Las anotaciones son iguales, por lo tanto se consideran consistentes. Figura G.1
2. No existen anotaciones sobre la sección, por lo tanto se considera una sección consistente. Figura G.2
3. Las anotaciones de una sección no son iguales, por lo tanto se consideran inconsistentes. Figura G.3

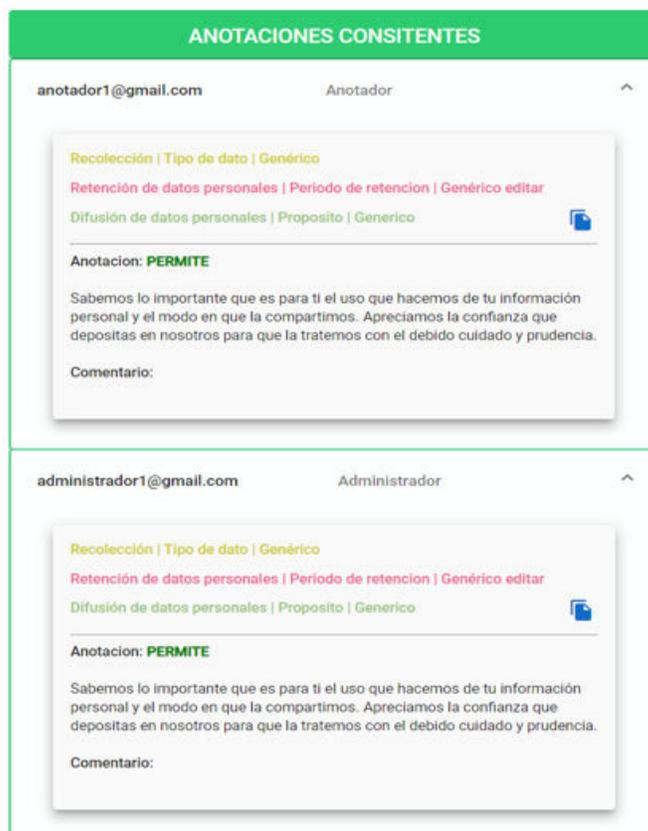


Figura G.1 Sección consistente (Anotaciones iguales)

ANOTACIONES CONSISTENTES		
anotador1@gmail.com	Anotador	^
administrador1@gmail.com	Administrador	^

Sección consistente (Sección sin anotaciones)

ANOTACIONES INCONSISTENTES		
anotador1@gmail.com	Anotador	^
administrador1@gmail.com	Administrador	^

Recolección | Tipo de dato | Genérico

Anotacion: **PERMITE**

Amazon.com, Inc. participa en el marco del Escudo de Privacidad entre la Unión Europea y Estados Unidos y entre Suiza y Estados Unidos.

Comentario:

Figura G.2 Sección Inconsistentes

En la Figura 2.15, se presentó el diagrama de flujo del cálculo de sección inconsistente en consolidación.

Finalmente, en el código siguiente se presenta lo que se implementó en el *backend* para el cálculo del coeficiente.

```

285 def consultar_inconsistencia(politica_id, secuencia, usuarios):
286     anotaciones_tuplas = []
287     for usuario in usuarios:
288         anotaciones_parrafo = (db.session.query(Anotacion, AnotacionValorRelacion)
289             .outerjoin(AnotacionValorRelacion, Anotacion.id == AnotacionValorRelacion.anotacion_id)
290             .outerjoin(Valor, AnotacionValorRelacion.valor_id == Valor.id)
291             .outerjoin(Atributo, Valor.atributo_id == Atributo.id)
292             .outerjoin(Tratamiento, Atributo.tratamiento_id == Tratamiento.id)
293             .outerjoin(Parrafo, Anotacion.parrafo_id == Parrafo.id)
294             .filter(Parrafo.politica_id == politica_id,
295                 Parrafo.secuencia == secuencia,
296                 Anotacion.usuario_id == usuario[1].id,
297                 Anotacion.consolidar == False)
298             .order_by(AnotacionValorRelacion.valor_id).all())
299         i = 0
300         anotacion_usuario = []
301         for anotacion in anotaciones_parrafo:
302             tupla = (anotacion[1].valor_id, anotacion[0].permite)
303             anotacion_usuario.insert(i, tupla)
304             i += 1
305         anotaciones_tuplas.append(anotacion_usuario)
306
307     inconsistencia = not all(x == anotaciones_tuplas[0] for x in anotaciones_tuplas)
308
309     return inconsistencia

```

Figura G.3 Cálculo de sección inconsistente en consolidación

ANEXO H

Código *frontend* de la herramienta de anotación y manual de usuario.

El código y el manual de usuario está disponible en el anexo digital

ANEXO I

Código *frontend* de la herramienta de visualización y manual de usuario.

El código y el manual de usuario está disponible en el anexo digital

ANEXO J

Código *backend* de la herramienta.

El código está disponible en el anexo digital

ANEXO K

Este anexo contiene la evidencia para los criterios de aceptación de distintas pruebas de funcionamiento.

Administración de roles usuarios en la herramienta HU-02

Como administrador de la herramienta requiero estar en la capacidad de:

1. Administrar roles de usuario de usuario haciendo las llamadas a las respectivas APIs RESTFul.

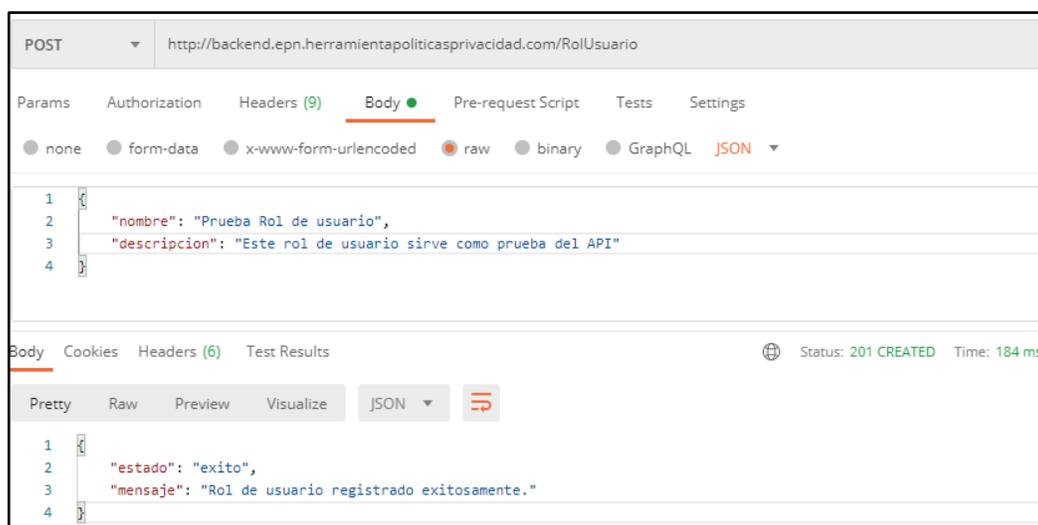


Figura K.1 Prueba de creación de un Rol de Usuario a través de *Postman*

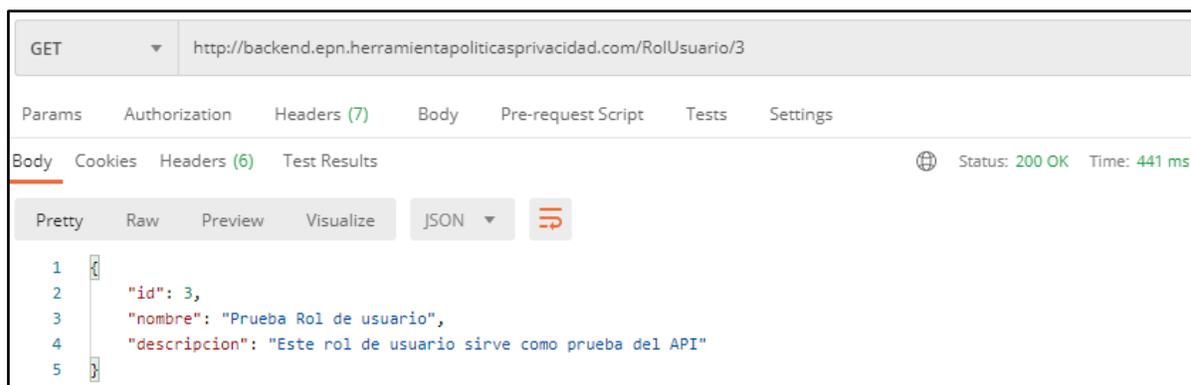


Figura K.2 Prueba de consulta de un Rol de Usuario a través de *Postman*

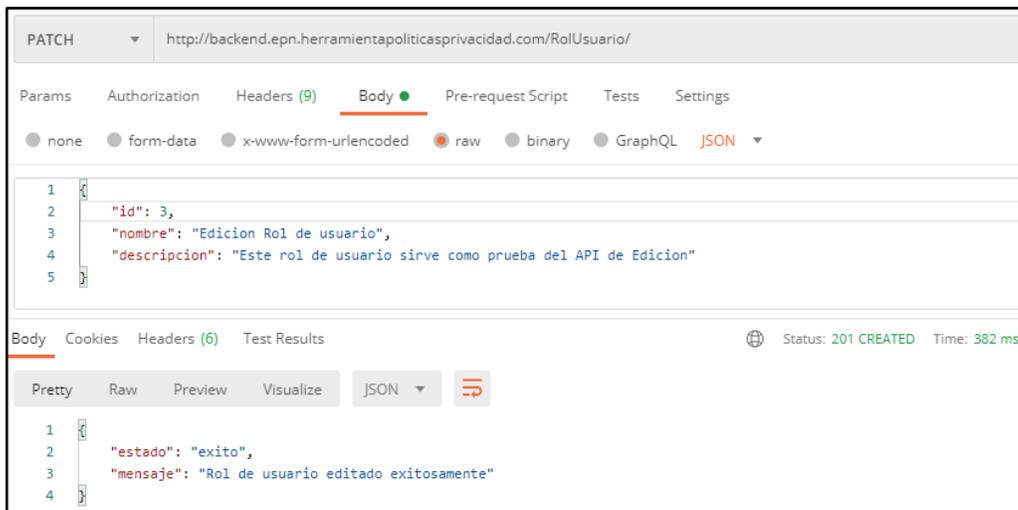


Figura K.3 Prueba de edición de un Rol de Usuario a través de *Postman*

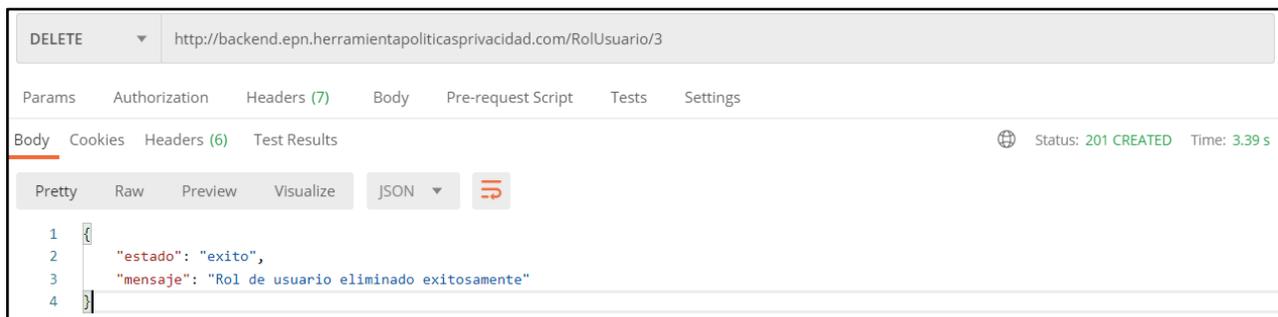


Figura K.4 Prueba de eliminación de un Rol de Usuario a través de *Postman*

Administración de colores tratamientos de datos HU-03

Como administrador de la herramienta requiero estar en la capacidad de:

1. Administrar los colores que serán asignados a los tratamientos de datos haciendo llamada a los respectivos APIs RESTFul.

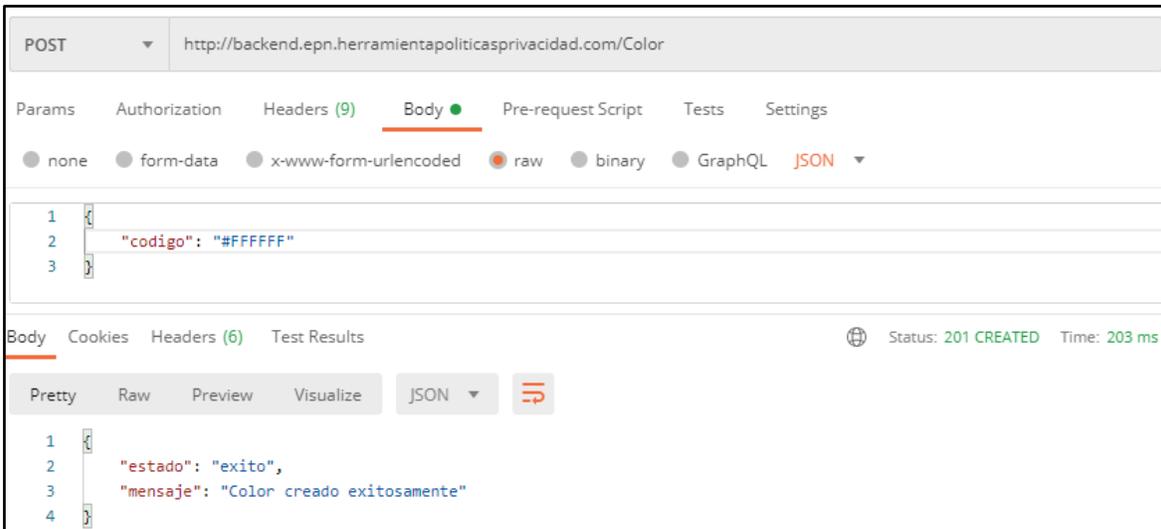


Figura K.5 Prueba de creación de color a través de Postman

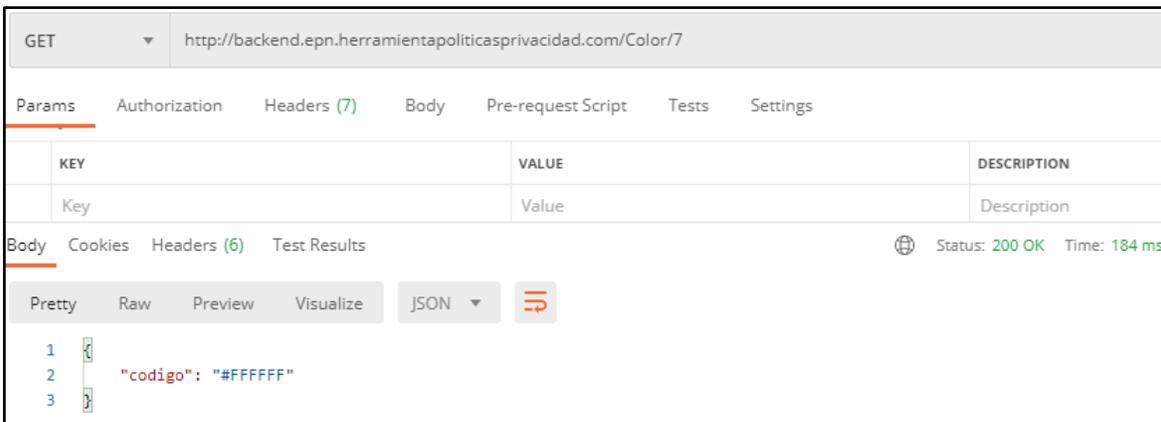


Figura K.6 Prueba de consulta de color a través de Postman

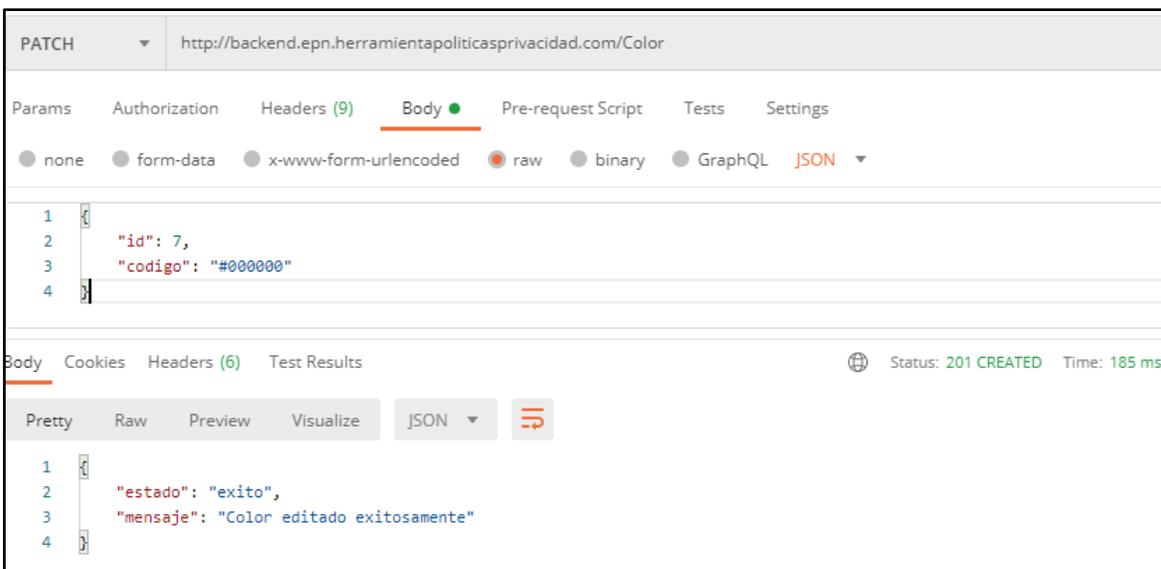


Figura K.7 Prueba de edición de color a través de Postman



Figura K.8 Prueba de eliminación de color a través de *Postman*

Administración de Atributos HU-04

Como administrador de la herramienta requiero estar en la capacidad de:

1. Crear un atributo y relacionarlo a un tratamiento de datos. Un atributo tendrá: descripción.

Figura K.9 Diálogo de creación de Atributo

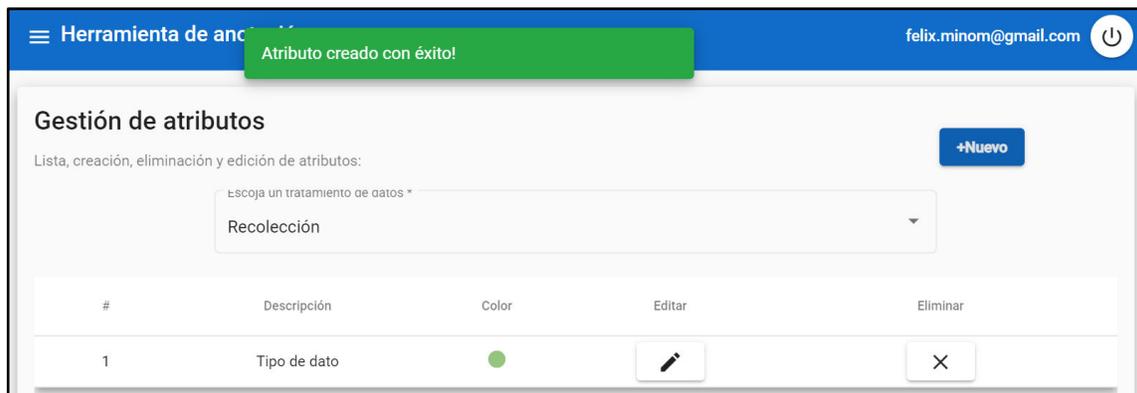


Figura K.10 Lista de atributos luego de creación

2. Visualizar los atributos pertenecientes a un tratamiento de datos.

Se guardó los atributos presentado en la Tabla 1.3 para el tratamiento “Recolección”.

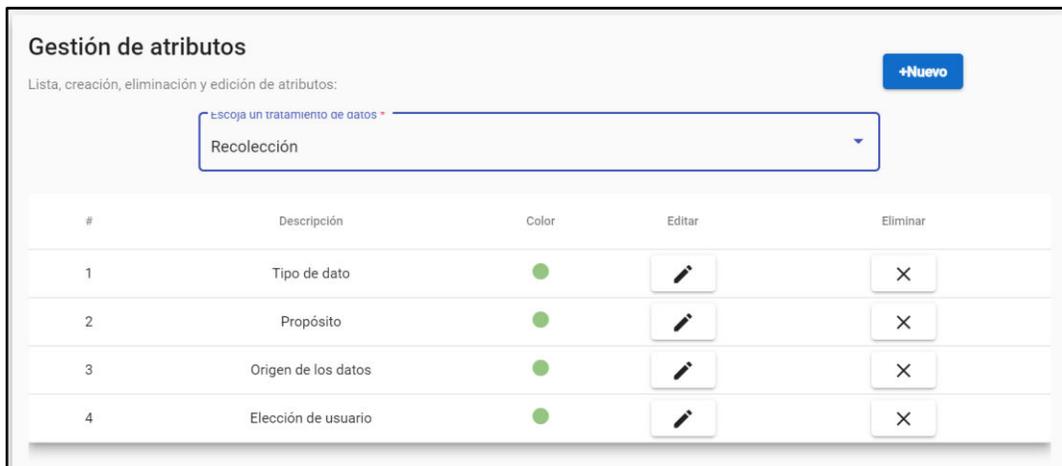


Figura K.11 Lista de atributos del tratamiento “Recolección”

3. Editar un atributo: descripción.



Figura K.12 Diálogo edición Atributo

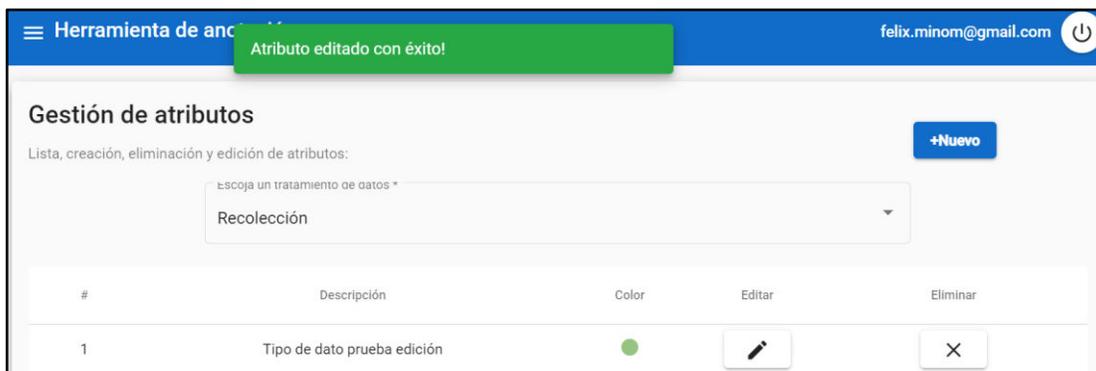


Figura K.13 Listado de atributos luego de edición de atributo

4. Eliminar un atributo. Manteniendo la integridad referencial.

Para probar que se respeta la integridad referencial fue necesario la creación de un valor para este atributo. Una vez creado se intentó eliminar el atributo y se obtuvo el error mostrado en la figura a continuación.

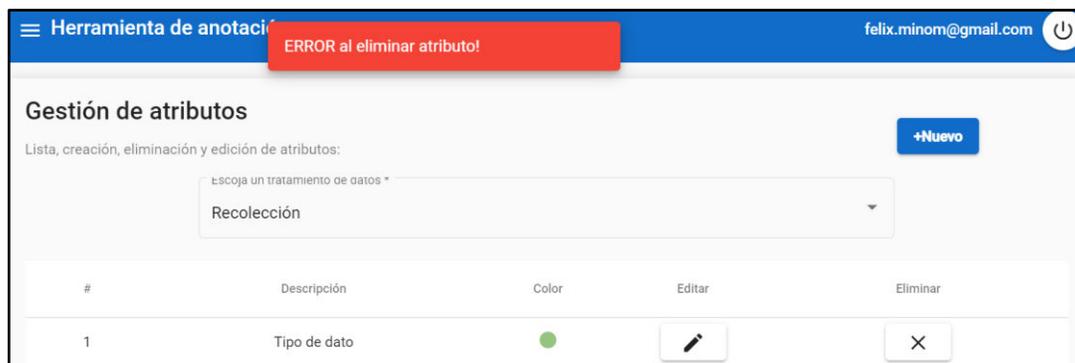


Figura K.14 Prueba de eliminación de atributo que contiene un valor

Una vez comprobado este funcionamiento el valor fue eliminado, se ejecutó la eliminación del atributo nuevamente y se obtuvo el siguiente resultado:

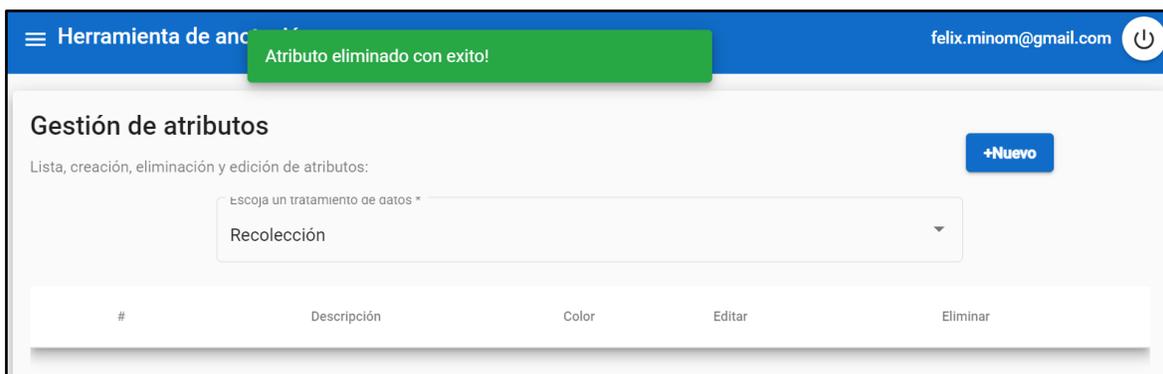


Figura K.15 Prueba de eliminación de atributo que no contiene valores.

Administración de Valores HU-05

Como administrador de la herramienta requiero estar en la capacidad de:

1. Crear un valor y relacionarlo a un atributo. Un valor tendrá: descripción.
2. Visualizar los valores pertenecientes a un atributo de un tratamiento de datos.

Creación de Valor

Nombre del valor

Guardar

Figura K.16 Diálogo de creación de Valor

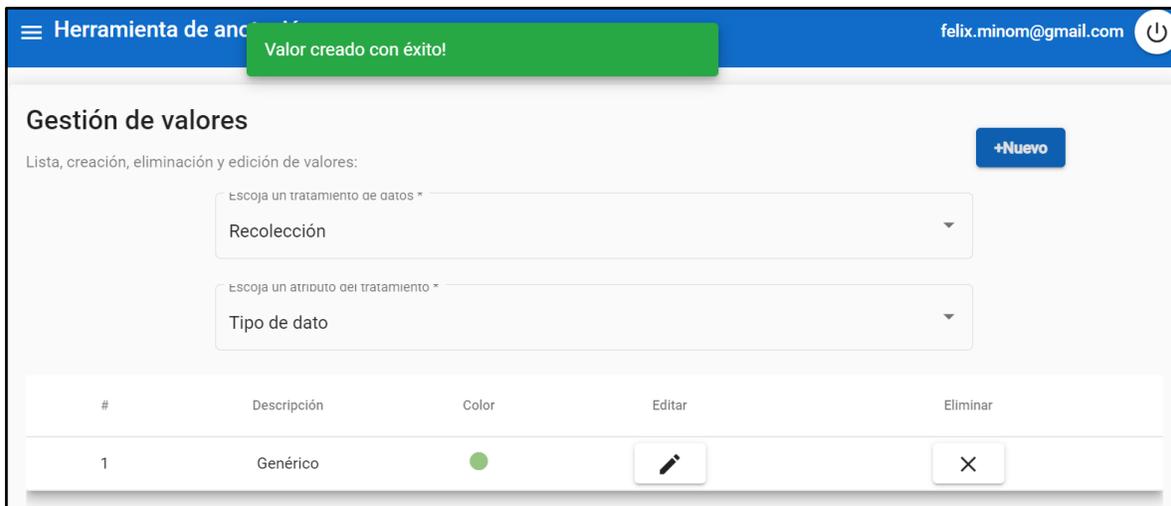


Figura K.17 Lista de valores luego de creación de valor

3. Editar un valor: descripción.

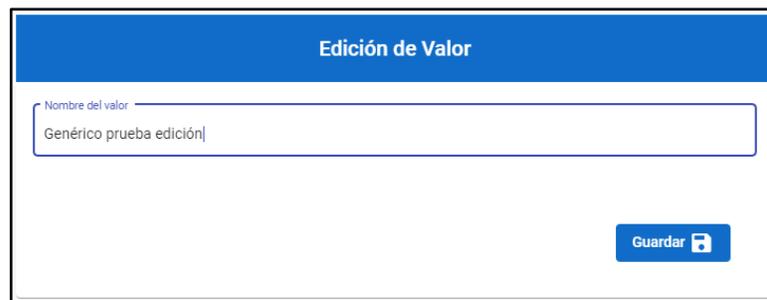


Figura K.18 Diálogo edición de Valor

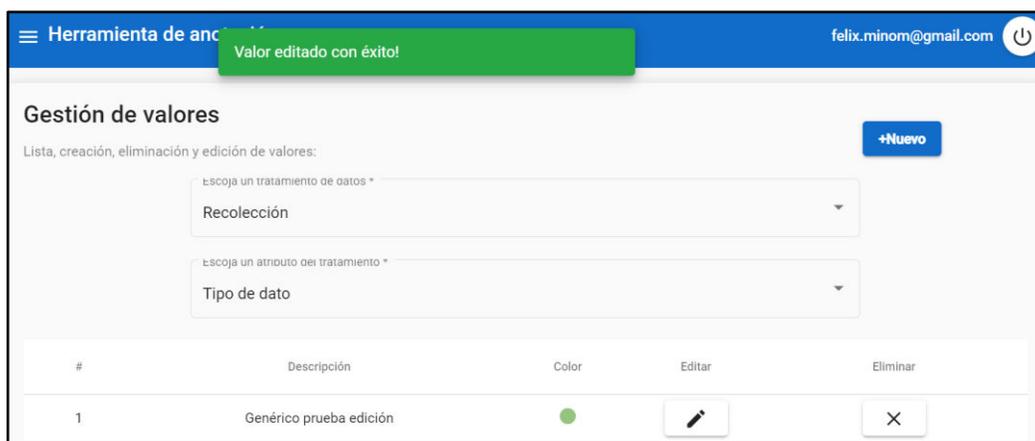


Figura K.19 Lista de valores luego de edición

4. Eliminar un valor. Manteniendo la integridad referencial.

Para realizar la prueba de integridad referencial se utilizó el valor creado para realizar una anotación. Al intentar eliminar el valor se obtiene el mensaje de error mostrado en la imagen a continuación.

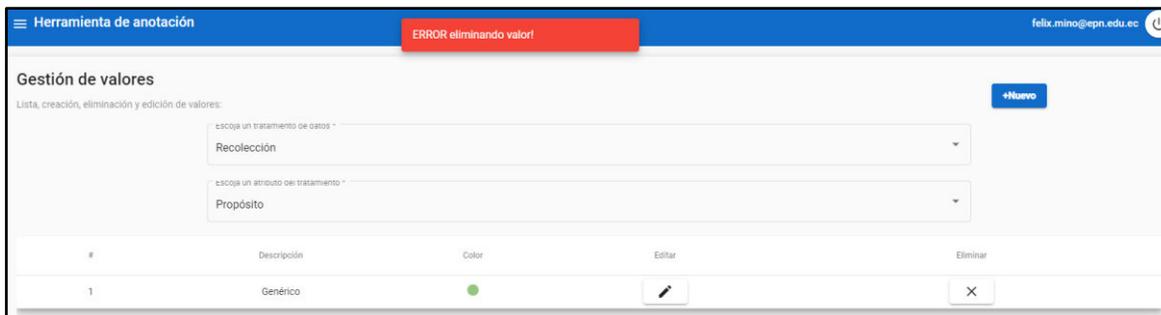


Figura K.20 Prueba de integridad referencial en eliminación de valor

Luego de realizar esta prueba se eliminó la anotación que usaba el valor. Se realizó la eliminación del valor “Genérico” del atributo “Tipo de dato” del tratamiento “Recolección” obteniendo el siguiente resultado:

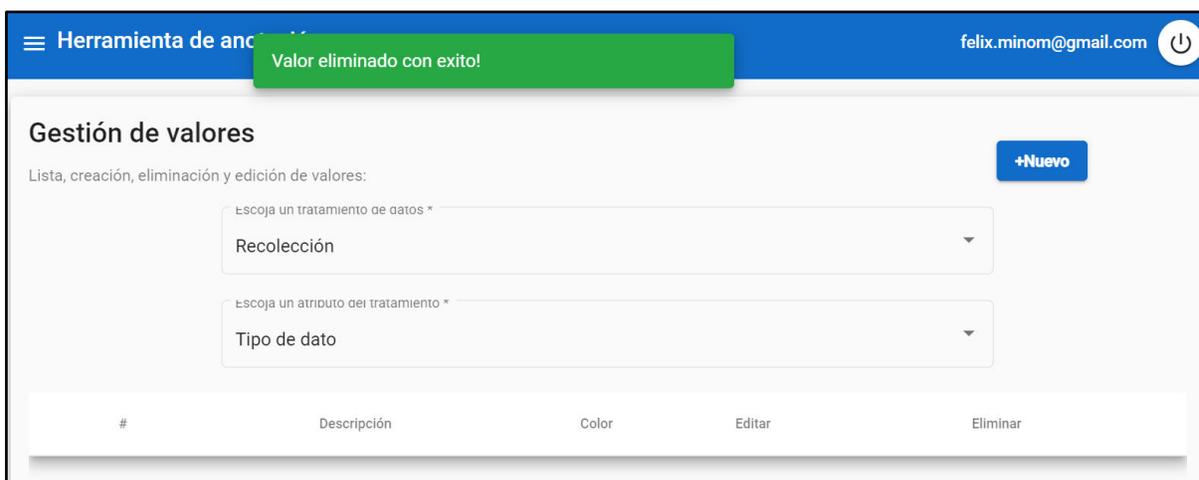


Figura K.21 Prueba de eliminación de valor

Administración de políticas de privacidad HU-06

1. Crear una política de privacidad. La política tendrá: nombre, URL de donde se obtuvo la política, fecha en la se obtuvo la política y un archivo en formato “txt”.

Subir política de privacidad

Nombre de la política *
Política de privacidad 1

URL donde se obtuvo la política *
https://www.lipsum.com/feed/html

Fecha en la que se obtuvo la política *
2020-11-26

Escoja un archivo
PoliticaPrivacidad1.txt

Cargar Política

Figura K.22 Diálogo creación de Política de Privacidad.

2. Dar formato al texto presente en el archivo “txt”. El carácter “+” precede a un título y dos saltos de línea representan una nueva sección.

Los archivos de las políticas de datos usadas para este caso de prueba se presentan en el Anexo E.

3. Previsualizar la política de privacidad que será guardada.

Al presionar el botón “Cargar Política” que se observa en la imagen anterior, se despliega el siguiente diálogo de la figura a continuación.

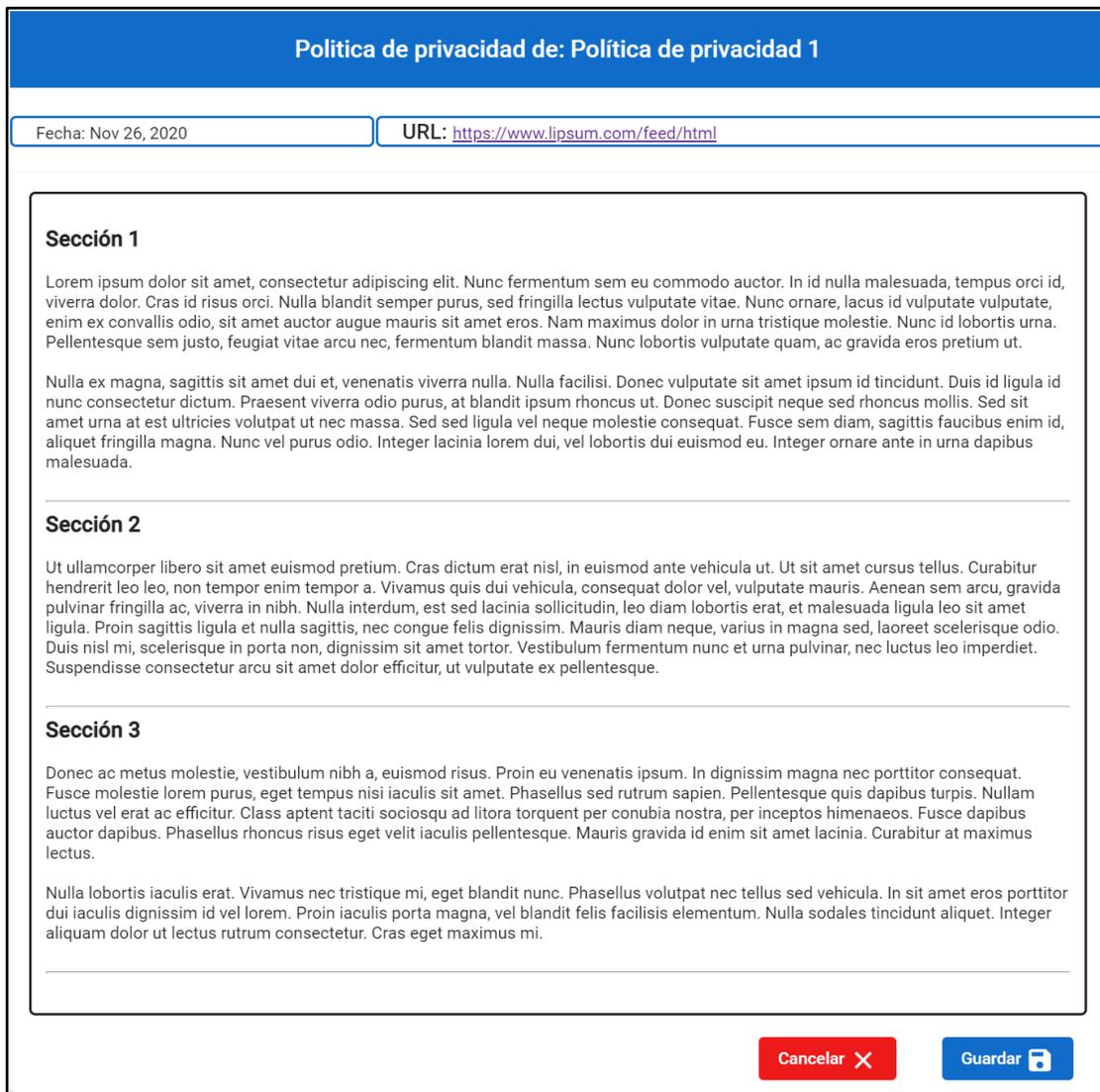


Figura K.23 Política de privacidad en diálogo de previsualización

4. Visualizar las políticas que fueron creadas.

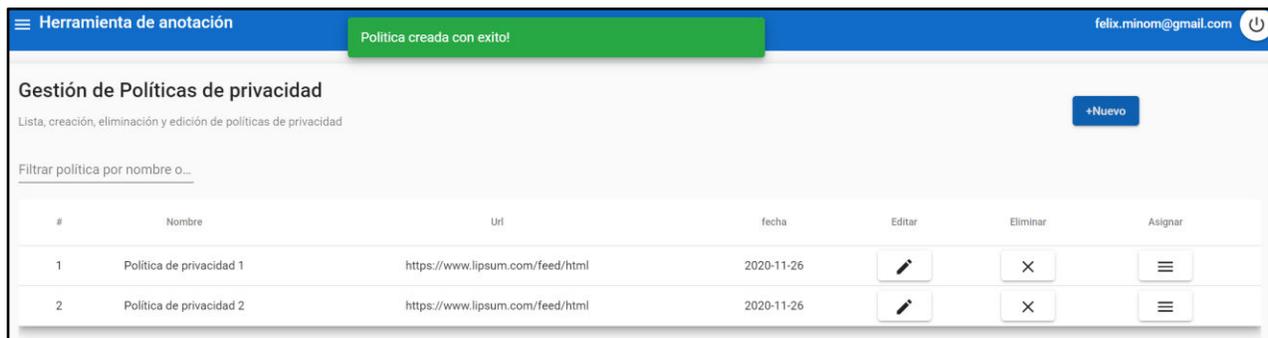


Figura K.24 Lista de políticas de privacidad creadas para el caso de prueba

5. Filtrar una política de privacidad por nombre.

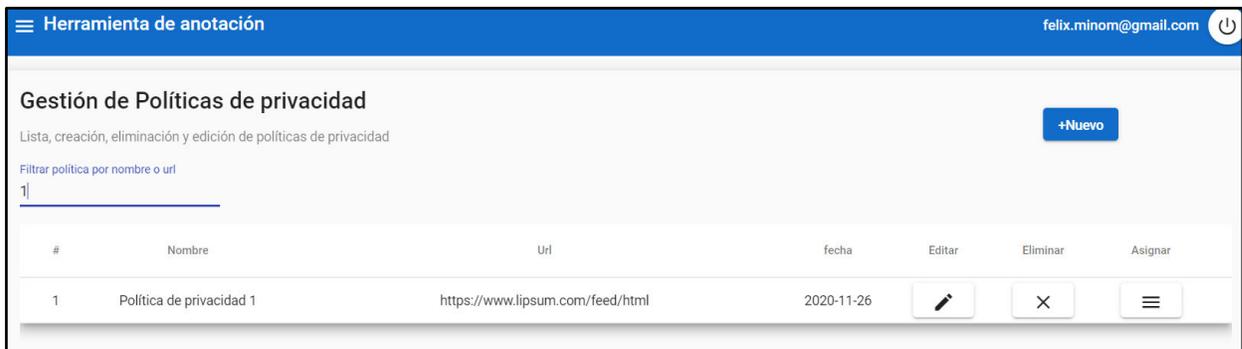


Figura K.25 Lista de políticas filtradas por texto "1"

6. Asignar una política a: 1) mínimo dos anotadores y 2) un administrador quien será el que realice la consolidación.

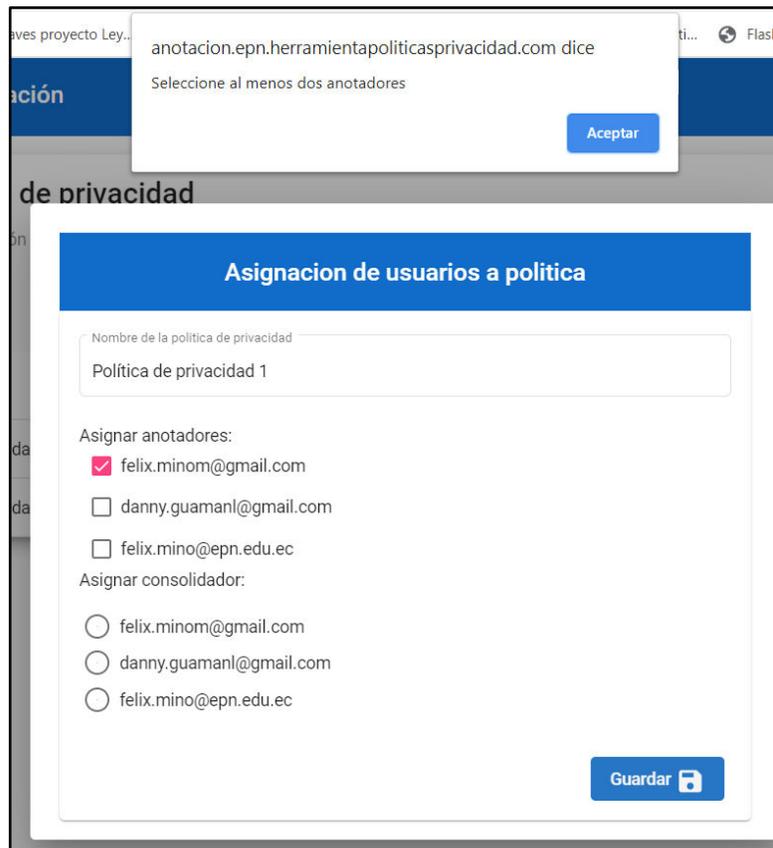


Figura K.26 Diálogo de asignación anotadores con error por número de anotadores

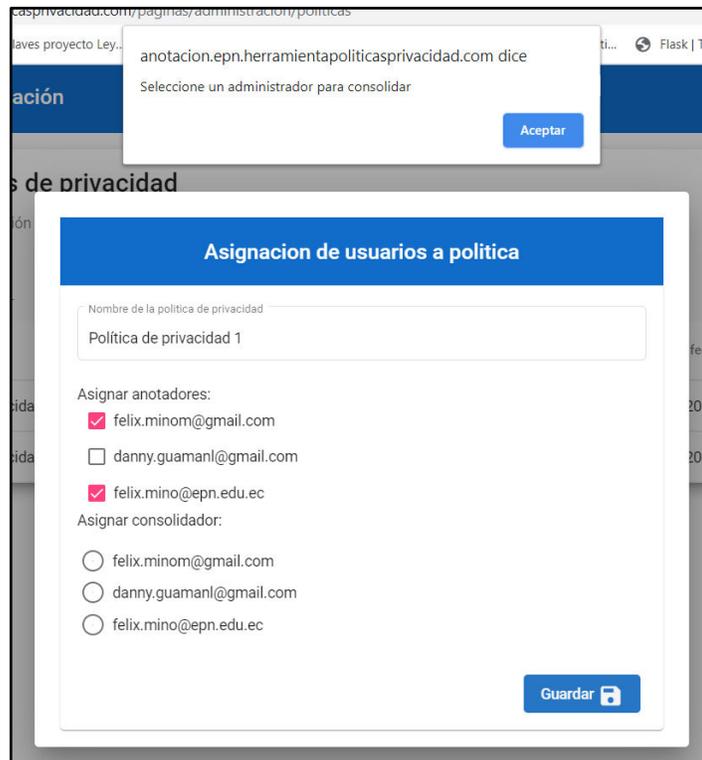


Figura K.27 Diálogo de asignación anotadores con error por ausencia de anotador

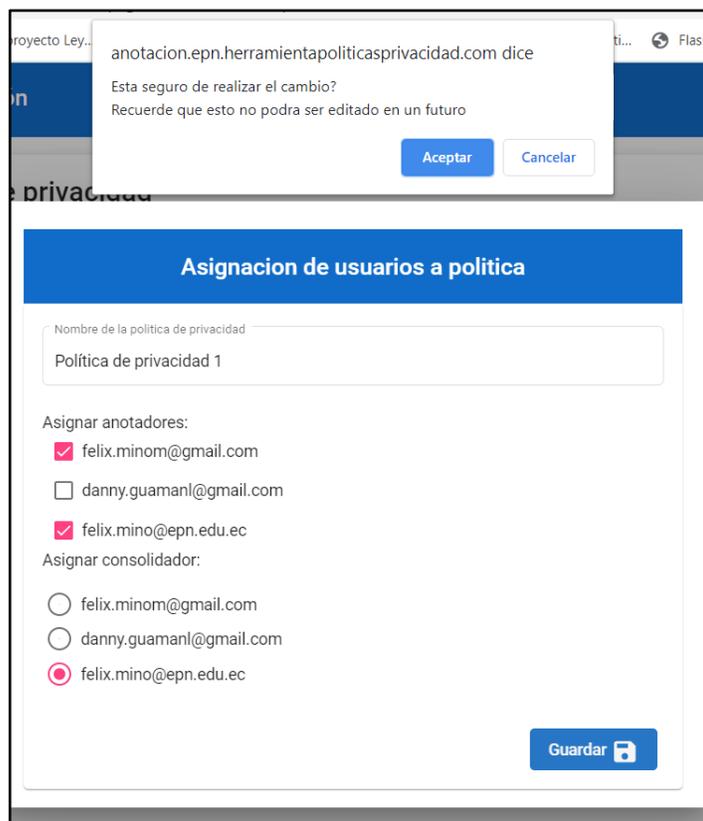


Figura K.28 Diálogo de asignación sin errores

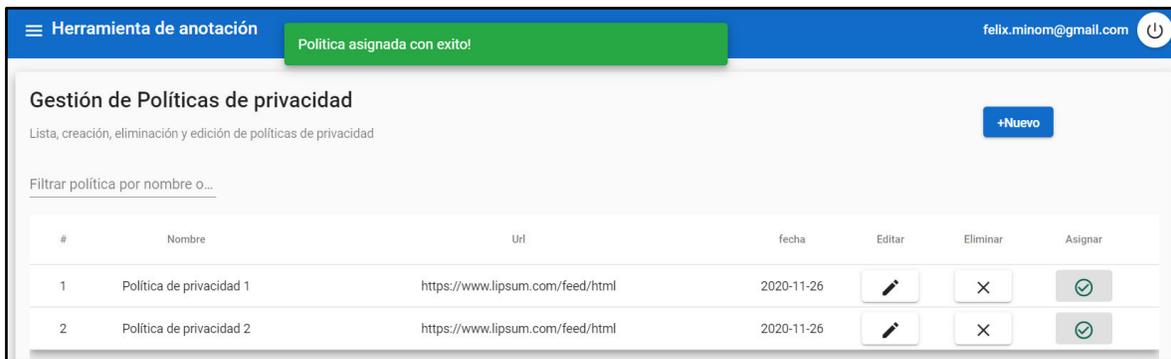


Figura K.29 Lista de políticas luego de asignación exitosa

7. Editar una política de privacidad: nombre, URL y fecha.



Figura K.30 Dialogo edición de Política de privacidad



Figura K.31 Lista de políticas de privacidad luego de edición

8. Eliminar una política de privacidad. Manteniendo la integridad referencial.

La política de privacidad no puede ser eliminada si tiene usuarios asignados o anotaciones en ella. A continuación, se presenta el diálogo de error al intentar eliminar una política de privacidad con usuarios asignados.

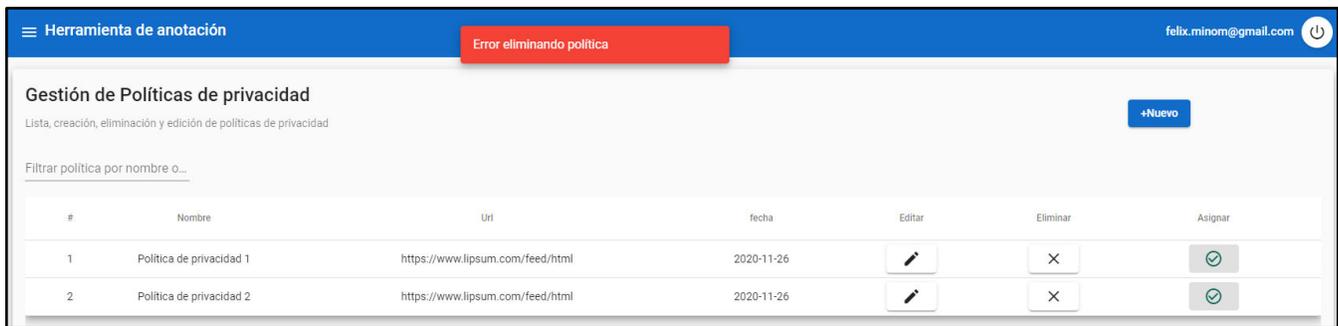


Figura K.32 Prueba de eliminación de política de privacidad con usuarios asignados

Consolidación de una política de privacidad HU-13

Como administrador de la herramienta requiero estar en la capacidad de:

1. Al dar *click* en una política de privacidad en la sección de consolidación ser redirigido a la página para la consolidación de la misma.
2. Tener una lista de los tratamientos de datos, sus atributos y valores. Solo se presentarán los tratamientos que incluyan al menos un atributo y un valor.
3. Observar la política separada por las secciones que fueron predefinidas.

Al realizar *click* en una política de la lista "Políticas de privacidad por anotar" se obtiene la siguiente interfaz de usuario. Como se observa, la lista de tratamientos disponibles está presente, así también la política de privacidad se encuentra separada por secciones.

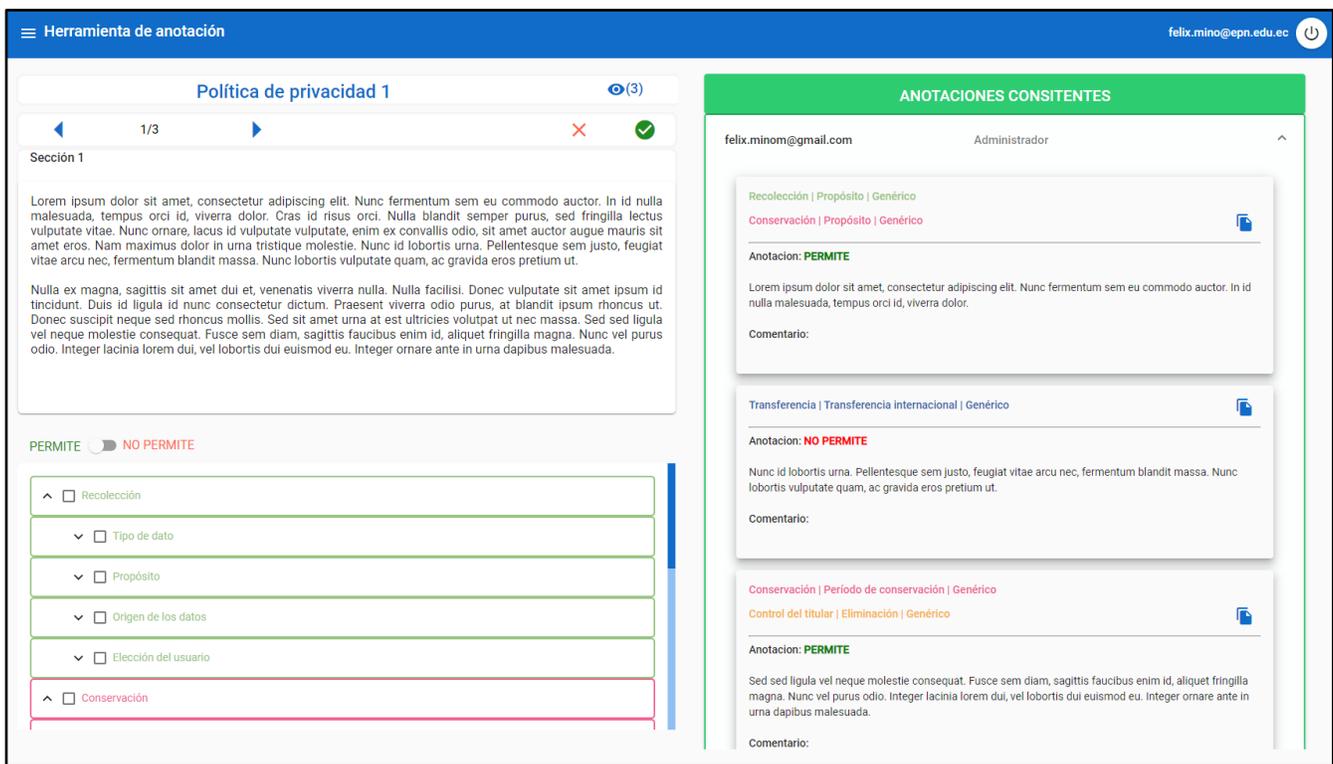


Figura K.33 Interfaz de consolidación de política de privacidad

4. Guardar una anotación. Una anotación consiste en la combinación de: uno o más valores, una sección de texto, un valor que identifica si la anotación ejecuta o no las acciones presentadas en la política y un comentario (opcional), sin embargo, el comentario no será cubierto por esta historia de usuario.

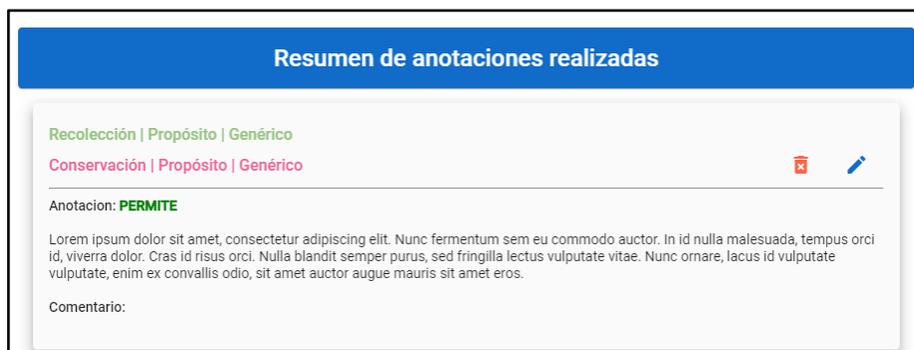


Figura K.34 Resumen de anotación en consolidación

5. Cuando se haya llegado a la última sección de la política y se de *click* en "siguiente" la herramienta debe notificar que se llegó al final del documento y requiero poder elegir si terminar o no la codificación.

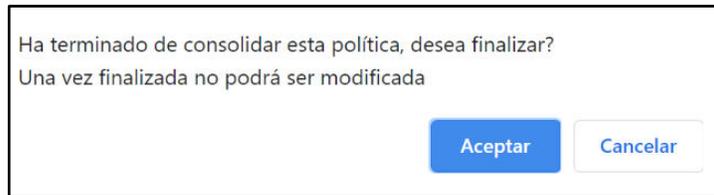


Figura K.35 Dialogo de notificación de finalización de política de privacidad

Visualización, edición y eliminación de anotaciones en consolidación HU-14

Como administrador de la herramienta requiero estar en la capacidad de:

1. Tener un conteo del número total de anotaciones que realicé sobre una sección de una política de privacidad en la etapa de consolidación.



Figura K.36 Conteo de anotaciones en consolidación

2. Visualizar las anotaciones que realicé sobre una sección en consolidación.

Al dar *click* en el contador de anotaciones se despliega el siguiente diálogo.



Figura K.37 Diálogo visualización de anotaciones en consolidación

3. Editar una anotación previamente realizada en consolidación. Se podrá editar: la acción (ejecuta o no ejecuta) y el comentario de una anotación. Los valores y el texto están fuera del alcance.

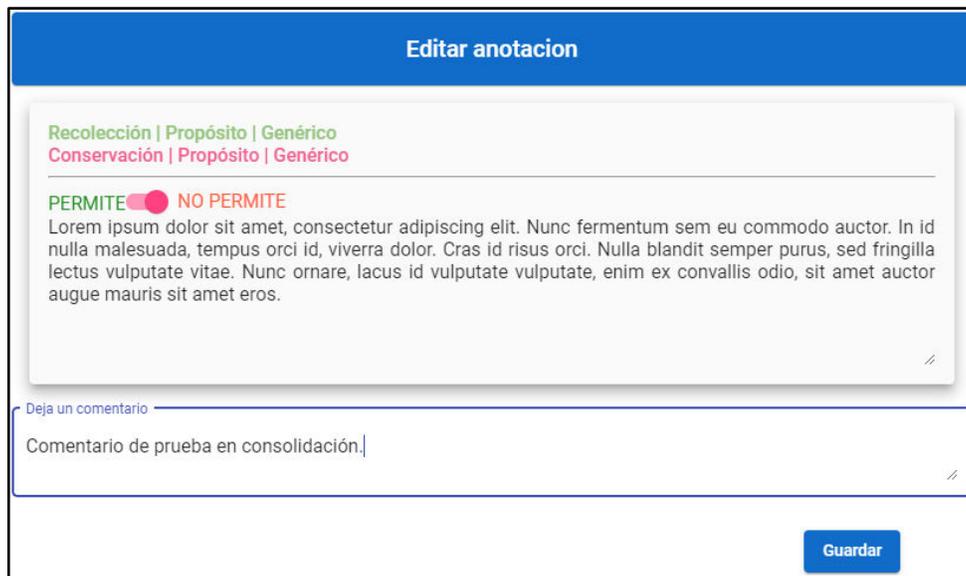


Figura K.38 Prueba de edición de anotación en consolidación (añadir comentario)

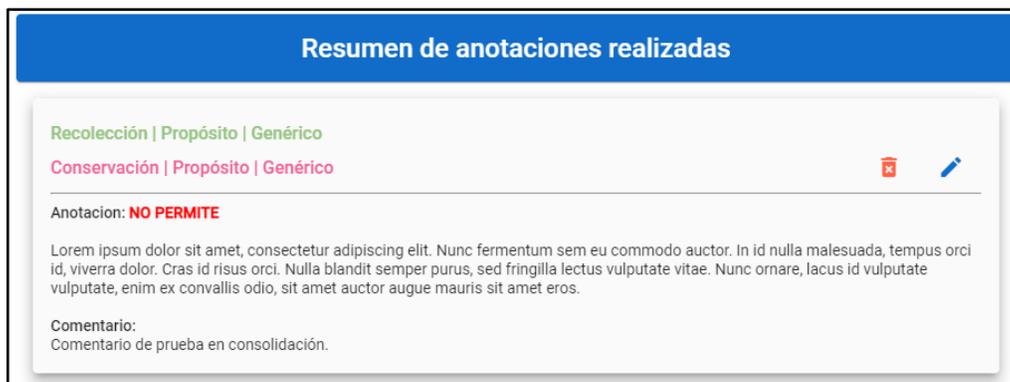


Figura K.39 Resumen de anotaciones luego de edición

4. Eliminar una anotación que realicé en consolidación.

Se realizó una anotación de manera aleatoria y como se observa en la figura a continuación, el conteo de anotaciones está incrementado.



Figura K.40 Conteo de anotaciones en consolidación luego de anotación aleatoria

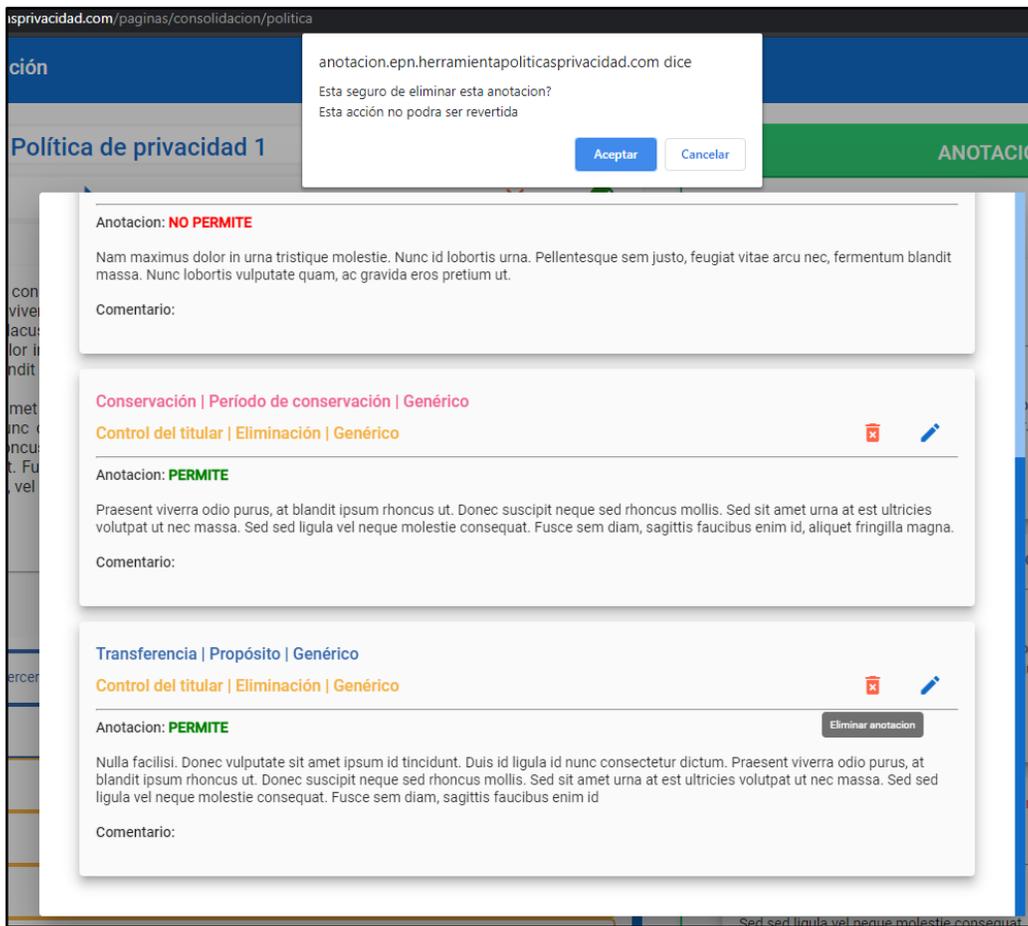


Figura K.41 Diálogo de confirmación de eliminación de anotación



Figura K.42 Conteo de anotaciones en consolidación luego de eliminación

ANEXO L

Cálculo manual de coeficiente inter-annotador de “Política de privacidad 2”

La Tabla L.1 contiene las anotaciones realizadas sobre “Política de privacidad 2”

Tabla L.1 Resumen de anotaciones sobre "Política de privacidad 2"

	Usuarios	Sec-ción	Anotación	Tratamiento	Atributo	Valor	Acción	
Política de privacidad 2	1*	1	2.1	Recolección	Origen de los datos	Genérico	No ejecuta	
				Transferencia	Transferencia internacional	Genérico		
			2.2 (=2.6)	Control del titular	Edición	Genérico	Ejecuta	
		2	Este párrafo no contendrá anotaciones					
		3	2.3 (=2.7)	Transferencia	Propósito	Genérico	Ejecuta	
			2.4	Conservación	Propósito	Genérico	Ejecuta	
	Control del titular	Edición		Genérico				
	2**	1	2.5	Recolección	Elección del usuario	Genérico	No ejecuta	
				Transferencia	Transferencia internacional	Genérico		
			2.6 (=2.2)	Control del titular	Edición	Genérico	Ejecuta	
		2	Este párrafo no contendrá anotaciones					
		3	2.7 (=2.3)	Transferencia	Propósito	Genérico	Ejecuta	
			2.8	Conservación	Propósito	Genérico	No ejecuta	
	Control del titular	Edición		Genérico				

La Tabla L.2 contiene las combinaciones únicas de tratamiento, atributo, valor y acción.

Tabla L.2 Combinaciones únicas de Tratamiento, atributo, valor y acción

#	Tratamiento	Atributo	Valor	Acción	¿Existe en usuario 1?	¿Existe en usuario 2?
1	Recolección	Origen de los datos	Genérico	No ejecuta	Si	No
2	Transferencia	Transferencia internacional	Genérico	No ejecuta	Si	Si
3	Control del titular	Edición	Genérico	Ejecuta	Si	Si
4	Transferencia	Propósito	Genérico	Ejecuta	Si	Si
5	Conservación	Propósito	Genérico	Ejecuta	Si	No
6	Recolección	Elección del usuario	Genérico	No ejecuta	No	Si
7	Conservación	Propósito	Genérico	No ejecuta	No	Si
8	Control del titular	Edición	Genérico	No ejecuta	No	Si

La Tabla L.3 corresponde a la matriz de datos de fiabilidad. Los datos son llenados de acuerdo con la Tabla L.2.

Tabla L.3 Matriz de datos de fiabilidad

#	1	2	3	4	5	6	7	8	Párrafo 2
Anotador 1	1	1	1	1	1	0	0	0	0
Anotador 2	0	1	1	1	0	1	1	1	0

En la sección 2.4.3.1 se presentó la matriz de coincidencia binaria, la cual corresponde a la Figura L.1.

Valores:	0	1	
0	O_{00}	O_{01}	n_0
1	O_{10}	O_{11}	n_1
Total de valores:	n_0	n_1	$n = x * N$

Figura L.1 Matriz de coincidencia binaria Alpha de Krippendorff

En la Figura 2 se presenta la matriz de coincidencia binaria correspondiente a "Política de privacidad 2" la cual es llenada con los datos de la Tabla 3.

	0	1	
0	2	5	$n_0 = 7$
1	5	6	$n_1 = 11$
	$n_0 = 7$	$n_1 = 11$	$n = 18$

Figura L.2 Matriz de coincidencia binaria de "Política de privacidad 2"

La ecuación L.1 corresponde a la ecuación simplificada para el cálculo del coeficiente Alpha de Krippendorff para datos binarios.

$$\alpha = 1 - (n - 1) \frac{O_{01}}{n_0 * n_1} \tag{L.1}$$

Reemplazando los valores de la Figura 2 en la fórmula anterior, se tiene:

$$\alpha = 1 - (18 - 1) \frac{5}{7 * 11} = -0.103389$$

ANEXO M

En este anexo se presenta la encuesta de satisfacción de los requisitos funcionales de la herramienta, así como las respuestas obtenidas a través de Google Forms.

Tabla M.1 Encuesta de satisfacción de requisitos funcionales

Encuesta de satisfacción herramientas de anotación y visualización			
La información proporcionada en esta encuesta será usada para la validación del proyecto: "DESARROLLO DE UNA HERRAMIENTA WEB PARA LA ANOTACIÓN DE TRATAMIENTOS DE DATOS PERSONALES EN POLÍTICAS DE PRIVACIDAD EN ESPAÑOL"			
Preguntas herramienta de anotación	Si	No	Parcialmente
P1: ¿Pudo iniciar y cerrar sesión?			
P2: ¿Pudo crear, visualizar, editar y eliminar los usuarios de la herramienta?			
P3: ¿Pudo crear, visualizar, editar y eliminar los tratamientos de datos?			
P4: ¿Pudo crear, visualizar, editar y eliminar los atributos de un tratamiento de datos?			
P5: ¿Pudo crear, visualizar, editar y eliminar los valores de un atributo?			
P6: ¿Pudo crear, visualizar, editar y eliminar las políticas de privacidad?			
P7: ¿Pudo visualizar las políticas de privacidad que le han sido asignadas para la etapa de anotación?			
P8: ¿Pudo anotar los tratamientos, atributos y valores de una política de privacidad?			
P9: ¿Pudo visualizar, editar y eliminar las anotaciones realizadas en la etapa de anotación?			
P10: Al estar en entrenamiento, ¿Fue notificado de inconsistencias en las anotaciones?			
P11: ¿Pudo visualizar las políticas de privacidad que le han sido asignadas para consolidar?			
P12: En la etapa de consolidación ¿Pudo observar el coeficiente interanotador de una política de privacidad?			
P13: ¿Pudo consolidar los tratamientos, atributos y valores de una política de privacidad?			
P14: ¿Pudo visualizar, editar y eliminar las anotaciones realizadas en la etapa de consolidación?			
P15: En la etapa de consolidación ¿Pudo visualizar las anotaciones realizadas en la etapa de anotación por los distintos usuarios (anotadores y administradores)?			

P16: En la etapa de consolidación ¿Fue notificado cuando una sección era inconsistente?			
Preguntas herramienta de visualización	Si	No	Parcialmente
P17: ¿Pudo observar la página de bienvenida?			
P18: ¿Pudo observar la lista de políticas de privacidad que han sido anotadas?			
P19: Al seleccionar una política de privacidad de la lista anterior ¿Pudo observar las anotaciones resaltadas?			
P20: ¿Pudo filtrar las anotaciones de una política de privacidad por el tratamiento de datos o por alguno de sus atributos?			
Preguntas generales			Respuesta
Responda en una escala del 1-5 donde 1 es 'En desacuerdo' y 5 'Totalmente de acuerdo'			
PG1: ¿Las herramientas de anotación y visualización son fiables en cuanto a los datos que presentan y almacenan?			
PG2: ¿Las interfaces de usuario son lo suficientemente intuitivas como para usar las herramientas sin necesidad de que una persona le guíe en su manejo?			
PG3: ¿Considera que las herramientas podrían usarse en experimentos reales?			
Responda en una escala del 1-5 donde 1 es 'No cumple las expectativas' y 5 'Cumple las expectativas'			
PG4: De manera general ¿Cómo calificaría a la herramienta de anotación?			
PG5: De manera general ¿Cómo calificaría a la herramienta de visualización?			
Preguntas de opinión			
PO1: ¿Qué mejoras considera que serían útiles una vez implementadas en las herramientas de anotación y visualización?			

A continuación se presentan las respuestas obtenidas desde Google Forms.

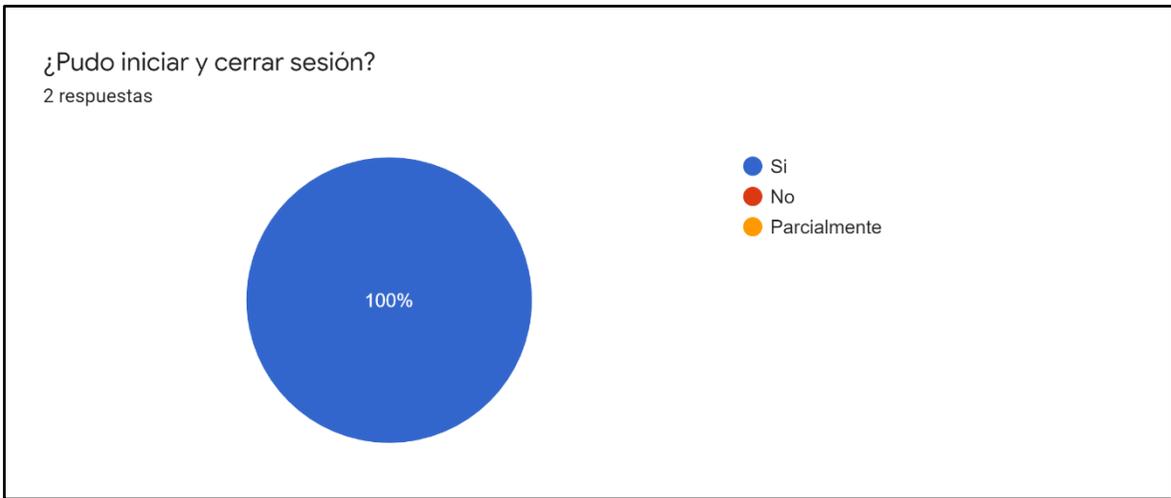


Figura M.1 Respuestas de la pregunta P1



Figura M.2 Respuestas de la pregunta P2



Figura M.3 Respuestas de la pregunta P3



Figura M.4 Respuestas de la pregunta P4



Figura M.5 Respuestas de la pregunta P5

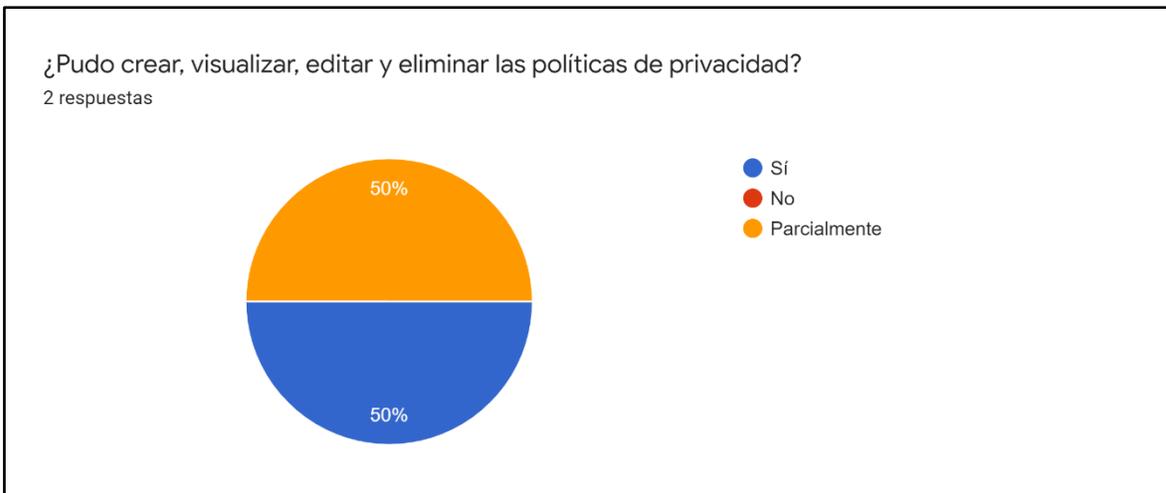


Figura M.6 Respuestas de la pregunta P6

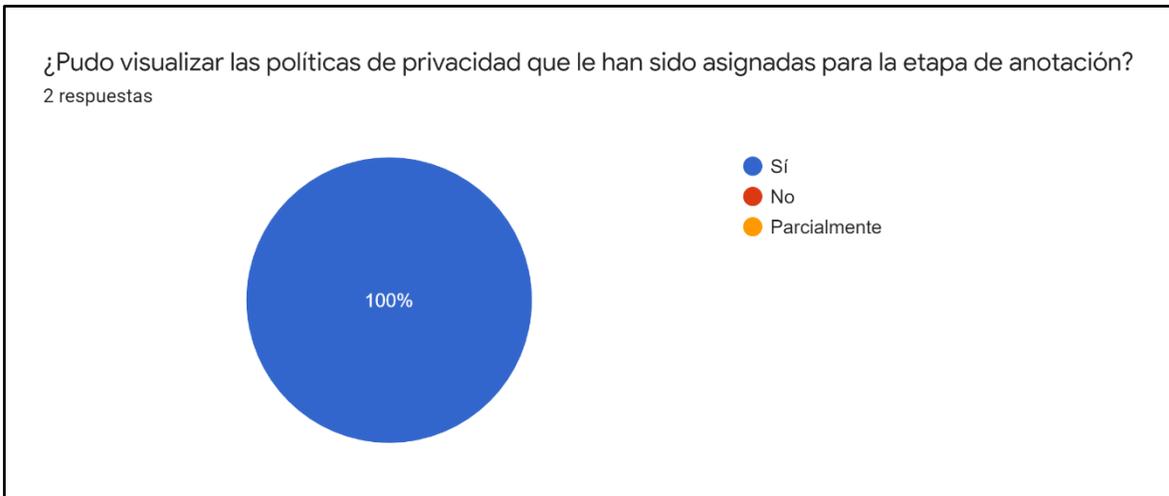


Figura M.7 Respuestas de la pregunta P7

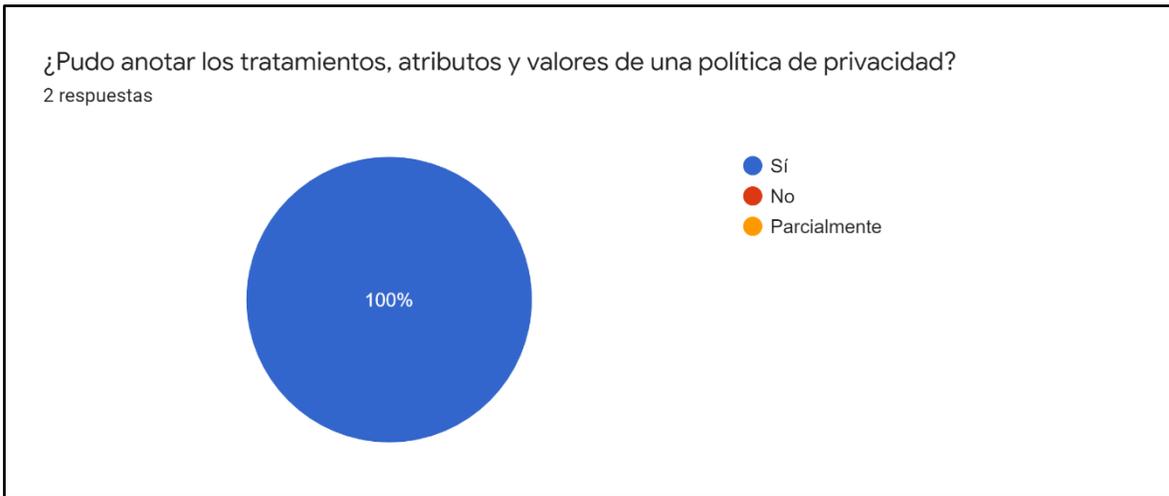


Figura M.8 Respuestas de la pregunta P8



Figura M.9 Respuestas de la pregunta P9

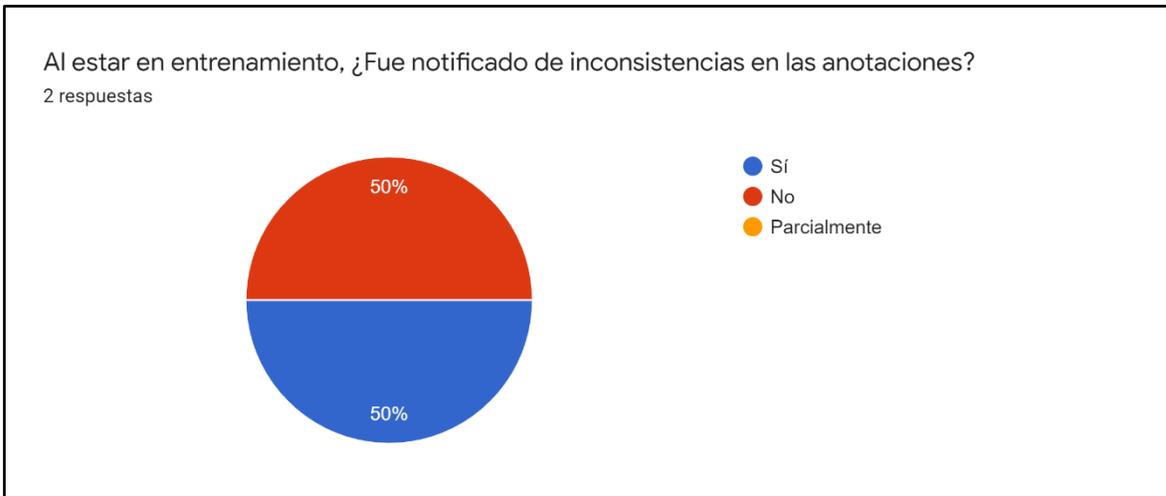


Figura M.10 Respuestas de la pregunta P10



Figura M.11 Respuestas de la pregunta P11

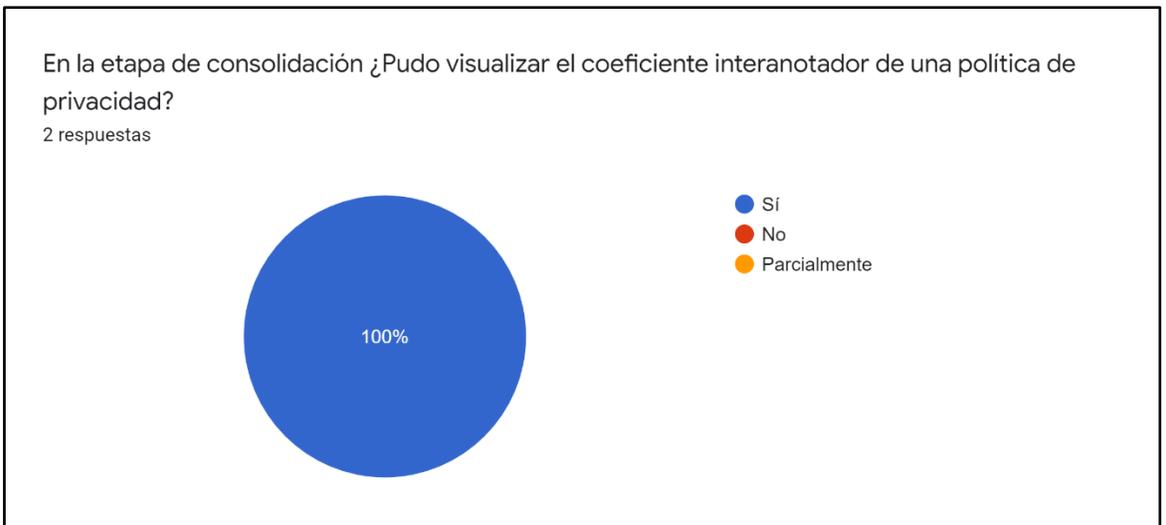


Figura M.12 Respuestas de la pregunta P12

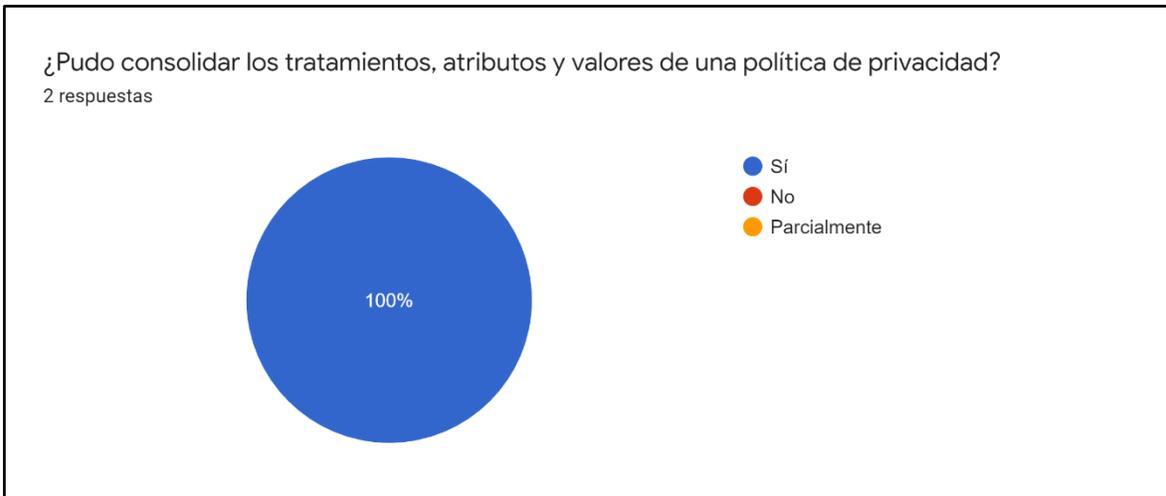


Figura M.13 Respuestas de la pregunta P13



Figura M.14 Respuestas de la pregunta P14

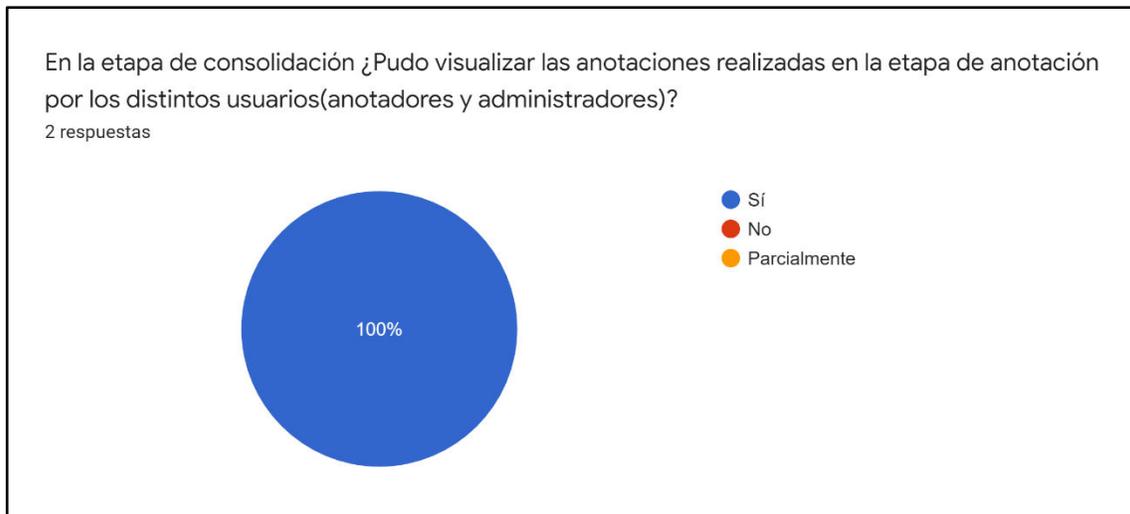


Figura M.15 Respuestas de la pregunta P15



Figura M.16 Respuestas de la pregunta P16

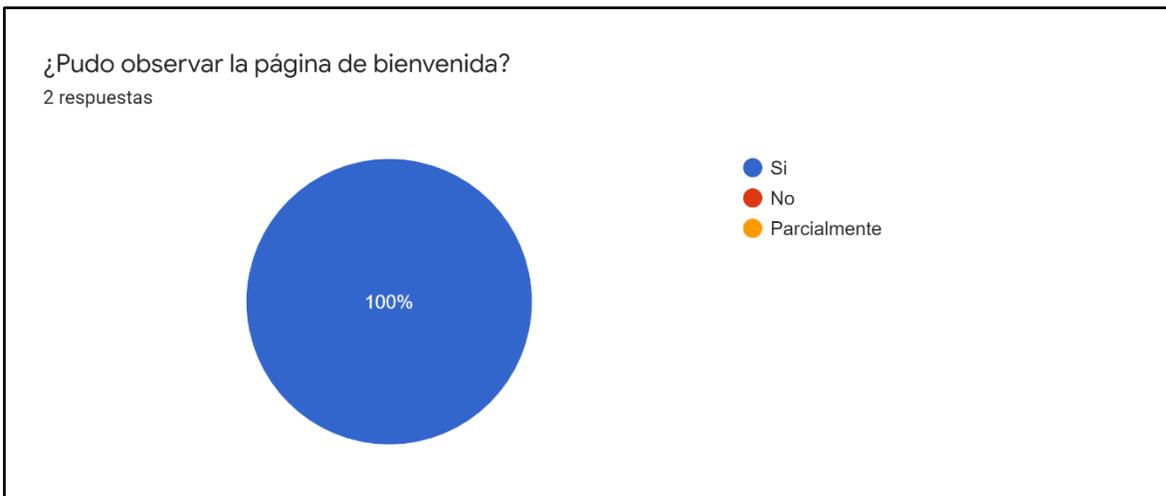


Figura M.17 Respuestas de la pregunta P17

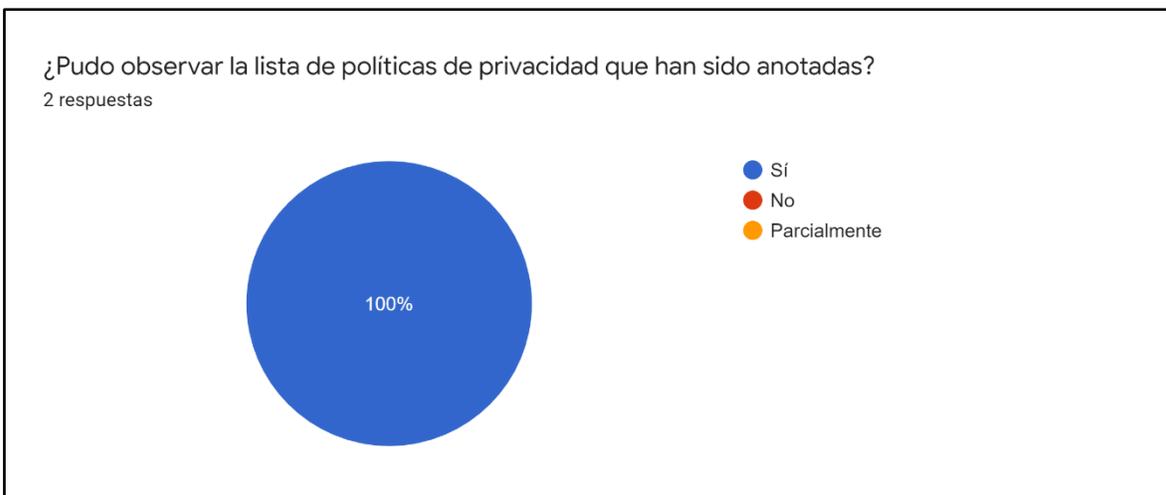


Figura M.18 Respuestas de la pregunta P18

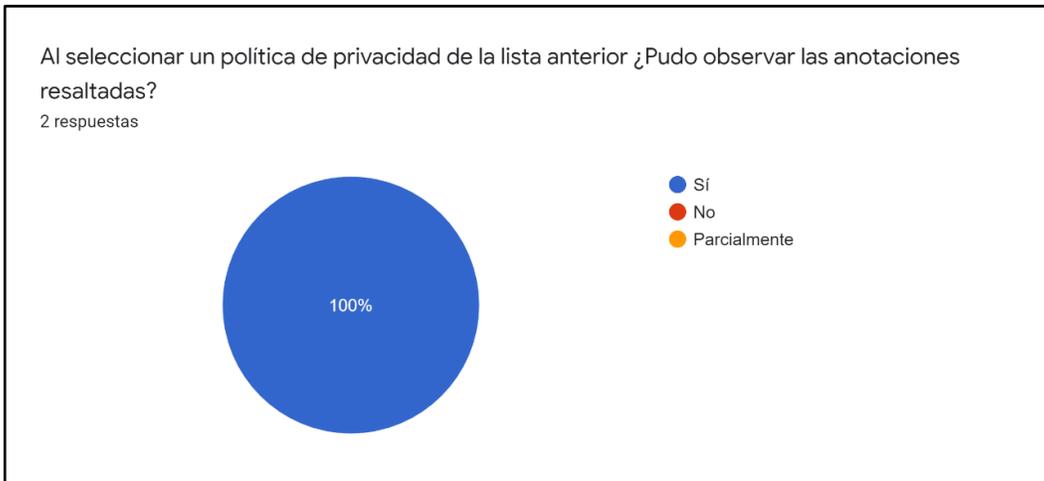


Figura M.19 Respuestas de la pregunta P19

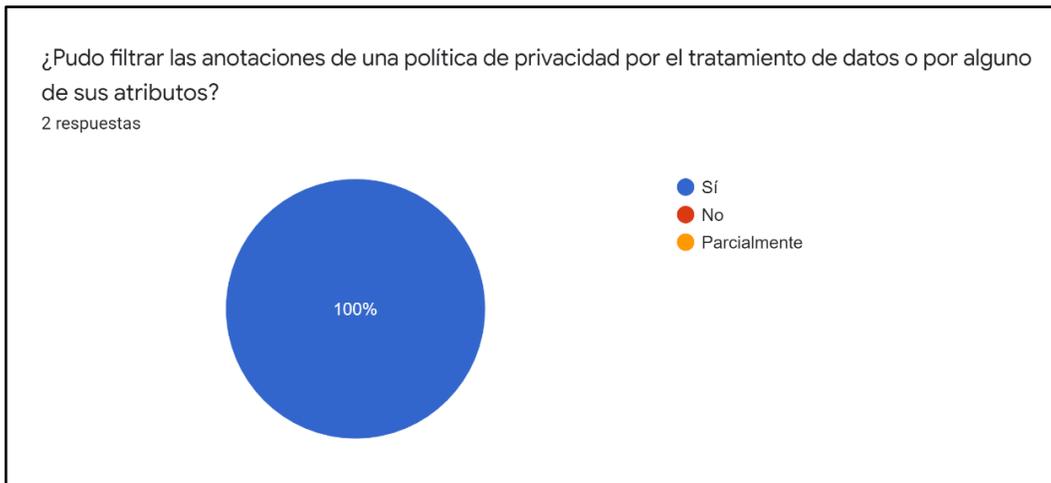


Figura M.20 Respuestas de la pregunta P20

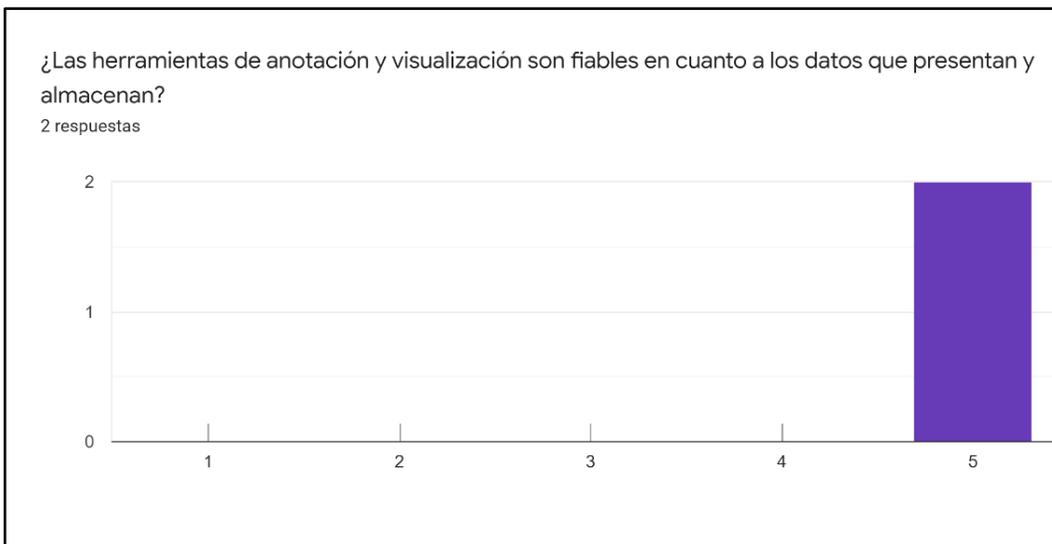


Figura M.21 Respuestas de la pregunta PG1

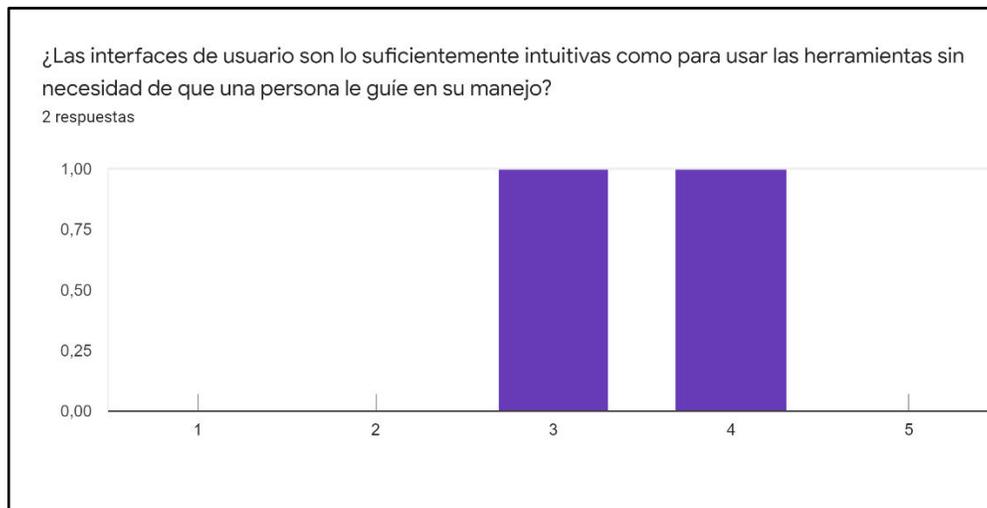


Figura M.22 Respuestas de la pregunta PG2



Figura M.23 Respuestas de la pregunta PG3

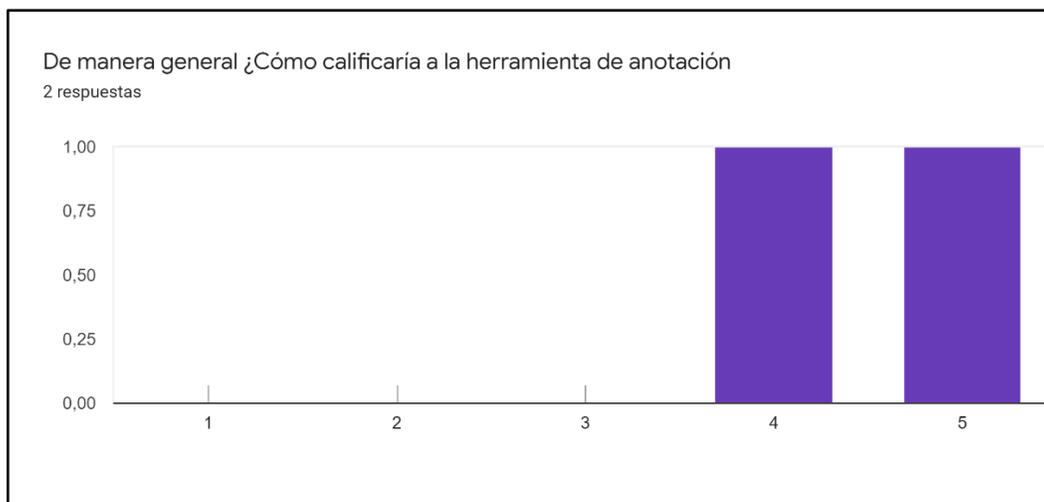


Figura M.24 Respuestas de la pregunta PG4

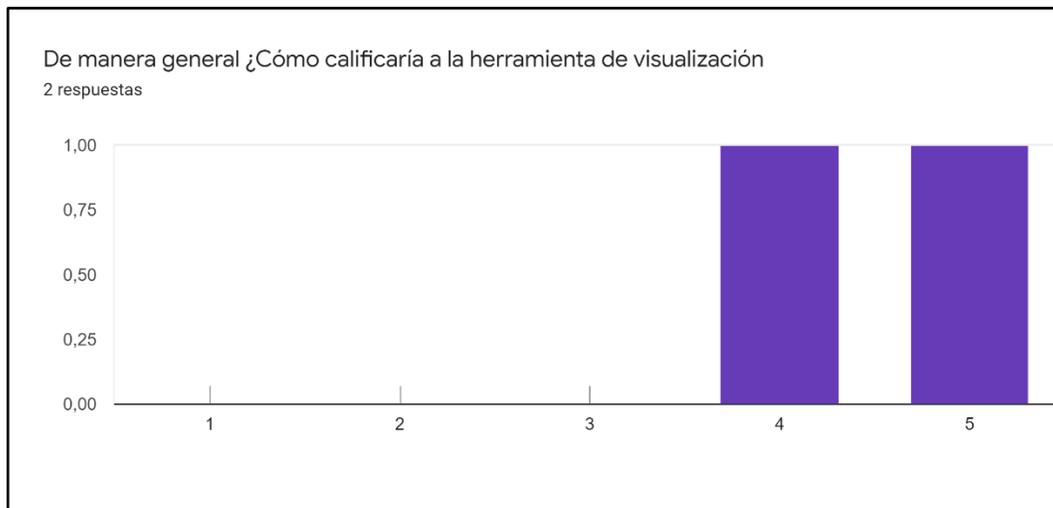


Figura M.25 Respuestas de la pregunta PG5

Opinión

¿Qué mejoras considera que serían útiles una vez implementadas en las herramientas de anotación y visualización?

2 respuestas

Herramienta de anotación:

- Permitir la configuración de atributos con la etiqueta de obligatorio y opcional.
- Permitir el cálculo del kalpha a nivel de segmento.
- Carga automática de textos de políticas a partir de un enlace (URL).
- Consolidación automática de segmentos de políticas consistentes.

Herramienta de visualización:

- Permitir la exportación de las anotaciones en formatos estructurados como YAML.
- Permitir el filtrado a nivel de valor.

- La adaptación a cualquier tamaño del navegador.
- Un lugar en el que cada anotador pueda ver las anotaciones que ha hecho de una política aunque no sean las que se han aprobado en consolidación.

Figura M.26 Respuestas de la pregunta PO1

ORDEN DE EMPASTADO