

ESCUELA POLÍTECNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**EVALUACIÓN DE PRÁCTICAS DE PRIVACIDAD EN
APLICACIONES MÓVILES**

**MEJORA DE LA HERRAMIENTA WEB PARA
ANOTACIÓN DE PRÁCTICAS DE PRIVACIDAD EN
POLÍTICAS TEXTUALES**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO
COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TECNOLOGÍAS DE INFORMACIÓN**

DAVID ALEXANDER PROAÑO RIVAS

david.proano01@epn.edu.ec

DIRECTOR: DANNY SANTIAGO GUAMÁN LOACHAMÍN

danny.guaman@epn.edu.ec

DMQ, ABRIL DE 2023

CERTIFICACIONES

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



DAVID ALEXANDER PROAÑO RIVAS

Certifico que el presente trabajo de integración curricular fue desarrollado por David Alexander Proaño Rivas, bajo mi supervisión.



DANNY SANTIAGO GUAMÁN LOACHAMÍN

DECLARACIÓN DE AUTORÍA

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

DAVID ALEXANDER PROAÑO RIVAS

DANNY SANTIAGO GUAMÁN LOACHAMÍN

DEDICATORIA

A mis padres Fausto y Flor, pido al cielo que los bendiga y me conceda la dicha de devolverles un poco de todo lo que me han dado.

Al amor de mi vida Andrea, quien estuvo apoyándome en cada paso, esa persona que tuvo paciencia y entrega para conmigo, gracias a ti hoy puedo dar un paso más.

A mi hija Emma, mi motor e inspiración para seguir adelante, entregaría todo por verte feliz, mi pequeña.

AGRADECIMIENTO

Agradezco a mis padres, por educarme con valores y darme ejemplo de trabajo constante. Doy gracias a la vida por ser hijo suyo y quiero prometerles que siempre estaré a su lado para devolverles todo su cariño, cuidado y atención.

Agradezco a mi esposa Andrea, por haber confiado en mí, y apoyarme durante mi vida académica, quiero que sepas amada mía, que jamás dejare de luchar para cumplir nuestros sueños y anhelos, te amo con todo mi corazón.

Agradezco al mejor regalo que la vida pudo darme, mi princesa Emma, por llegar a mi vida en el momento preciso y ayudarme a ser una mejor persona.

Agradezco a mis hermanas Christina y Paola, por su aliento, enseñanza, sabiduría y apoyo en los momentos difíciles hacia mí y mi familia.

Agradezco a mis suegros Joselito y Cesilia, por abrirme las puertas de su hogar y hacerme sentir como un hijo más.

Agradezco a la Escuela Politécnica Nacional, por darme la oportunidad de pertenecer a tan distinguida institución y formarme profesionalmente.

Agradezco a mi tutor, PhD. Danny Guamán por su guía durante el desarrollo de este trabajo.

Finalmente, agradezco a mis amigos que me dio la universidad Gabriel, Leo, Pancho, Bryan y Santa.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS.....	IX
ÍNDICE DE CÓDIGOS	X
RESUMEN.....	XI
ABSTRACT	XII
1 INTRODUCCIÓN	1
1.1 OBJETIVOS	2
1.1.1 OBJETIVO GENERAL	2
1.1.2 OBJETIVOS ESPECÍFICOS	2
1.2 ALCANCE.....	2
1.3 MARCO TEÓRICO	3
1.3.1 PRIVACIDAD Y PROTECCIÓN DE DATOS PERSONALES.....	3
1.3.2 TECNOLOGÍAS PARA PROTECCIÓN DE LA PRIVACIDAD	4
1.3.3 EVALUACIÓN DE CUMPLIMIENTO DE PRIVACIDAD Y PROTECCIÓN DE DATOS.....	5
1.3.4 ANNOTOOL-CLIIP - HERRAMIENTA PARA LA ANOTACIÓN DE TRATAMIENTOS DE DATOS PERSONALES.....	6
1.3.5 HERRAMIENTAS DE DESARROLLO UTILIZADAS EN EL TRABAJO	8
1.3.6 METODOLOGÍA ÁGIL SCRUM.....	9
2 METODOLOGIA	11
2.1 ANÁLISIS	11
2.1.1 REQUISITOS NO FUNCIONALES	12
2.1.2 REQUISITOS FUNCIONALES	12
2.1.3 HISTORIAS DE USUARIO	12
2.1.4 DIAGRAMA DE CASOS DE USO	16
2.1.5 PRODUCT BACKLOG.....	16
2.1.6 SPRINT BACKLOG.....	19
2.2 DISEÑO.....	21
2.2.1 DISEÑO SPRINT 1 – DESPLIEGUE DEL ENTORNO DE DESARROLLO	21

2.2.2	DISEÑO SPRINT 2 - EXTRACCIÓN DE POLITICAS	21
2.2.3	DISEÑO SPRINT 3 - GENERACIÓN DE REPORTES	25
2.2.4	DISEÑO SPRINT 4 – INTERNACIONALIZACIÓN	26
2.3	IMPLEMENTACIÓN	27
2.3.1	IMPLEMENTACIÓN SPRINT 1 – DESPLIEGUE DEL ENTORNO DE DESARROLLO	27
2.3.2	IMPLEMENTACIÓN SPRINT 2 - EXTRACCIÓN DE POLÍTICAS	28
2.3.3	IMPLEMENTACIÓN SPRINT 3 – REPORTES	39
2.3.4	IMPLEMENTACIÓN SPRINT 4 - INTERNACIONALIZACIÓN	50
2.4	VALIDACIÓN Y VERIFICACIÓN	53
2.4.1	PRUEBAS SPRINT 1 – DESPLIEGUE ENTORNO DE DESARROLLO.....	53
2.4.2	PRUEBAS SPRINT 2 – EXTRACCIÓN POLÍTICAS	54
2.4.3	PRUEBAS SPRINT 3 – REPORTES	58
2.4.4	PRUEBAS SPRINT 4 – INTERNACIONALIZACIÓN	63
2.4.5	RESUMEN.....	64
3	CONCLUSIONES Y RECOMENDACIONES.....	66
3.1	CONCLUSIONES	66
3.2	RECOMENDACIONES.....	67
4	REFERENCIAS	68
5	ANEXOS.....	70

ÍNDICE DE FIGURAS

Figura 1.1.	Funcionalidad de Herramienta de Anotación.....	2
Figura 1.2.	Método de desarrollo.....	3
Figura 1.3.	Despliegue de la Herramienta de Anotación.....	7
Figura 1.4.	Marco de trabajo Scrum [14].	9
Figura 2.1.	Diagrama de casos de uso de la herramienta de anotación.	16
Figura 2.2.	Arquitectura rediseñada de la herramienta.	22
Figura 2.3.	Diagrama de Actividades Políticas.	23
Figura 2.4.	Diseño Interfaz Bienvenida.	24
Figura 2.5.	Diseño Ayuda Extracción de Texto de Política de Privacidad.....	24
Figura 2.6.	Diseño Interfaz Políticas.	25
Figura 2.7.	Diagrama de Actividades Reportes.	25
Figura 2.8.	Diseño Interfaz Reportes.	26
Figura 2.9.	Diagrama de Actividades Internacionalización.....	27
Figura 2.10.	Diseño Interfaz Nueva Bienvenida.	27
Figura 2.11.	Chromedriver Actual.....	28
Figura 2.12.	Actualización de módulos.	30
Figura 2.13.	Interfaz Políticas.....	32
Figura 2.14.	Archivo de Texto.....	33
Figura 2.15.	Ingreso de Archivo de Texto.....	33
Figura 2.16.	Resultado de Leer Archivo de Texto.....	34
Figura 2.17.	Resultado de Extracción de Texto.....	38
Figura 2.18.	Ayuda Formato de Texto de Política de Privacidad.....	39
Figura 2.19.	Interfaz Reportes.	46
Figura 2.20.	Formulario Reportes.....	49
Figura 2.21.	Fichero de Reportes Generados.....	49
Figura 2.22.	Reportes YAML Generados.....	49
Figura 2.23.	Reporte CSV Generado.	49
Figura 2.24.	Archivos JSON.....	51
Figura 2.25.	Módulos Corregidos.....	53
Figura 2.26.	Interfaz Bienvenida Corregida.	53
Figura 2.27.	Archivo de Texto para Ingresar.....	54

Figura 2.28.	Selección de Archivo de Texto.	55
Figura 2.29.	Validación de Archivo de Texto.	55
Figura 2.30.	Proceso de Extracción de Texto.	55
Figura 2.31.	Resultado de Extracción.	56
Figura 2.32.	Formulario Políticas.	56
Figura 2.33.	Extracción Siguiete URL.	57
Figura 2.34.	Políticas de Privacidad Ingresadas.	57
Figura 2.35.	Anotaciones Prueba1.	59
Figura 2.36.	Anotaciones Prueba2.	59
Figura 2.37.	Herramienta sin Anotaciones.	60
Figura 2.38.	Herramienta con Anotaciones.	60
Figura 2.39.	Formulario Reportes.	61
Figura 2.40.	Generación de Reporte.	61
Figura 2.41.	Reportes Generados.	61
Figura 2.42.	Reporte CSV.	62
Figura 2.43.	Reporte YAML.	62
Figura 2.44.	Interfaz de Bienvenida.	63
Figura 2.45.	Interfaz Inglés.	63
Figura 2.46.	Interfaz Español.	64

ÍNDICE DE TABLAS

Tabla 1.1.	Roles y Módulos.	7
Tabla 2.1.	Roles Srum.....	11
Tabla 2.2.	Formato de Historia de Usuario.....	12
Tabla 2.3.	Historia de Usuario HU-01 – Adaptación de tecnologías.....	12
Tabla 2.4.	Historia de Usuario HU-02 – Compatibilidad de actualizaciones.	13
Tabla 2.5.	Historia de Usuario HU-03 – Despliegue de herramienta.	13
Tabla 2.6.	Historia de Usuario HU-04 – Internacionalización de la aplicación. ...	14
Tabla 2.7.	Historia de Usuario HU-05 – Extracción de políticas individuales.....	14
Tabla 2.8.	Historia de Usuario HU-06 – Extracción de políticas grupales.....	15
Tabla 2.9.	Historia de Usuario HU-07 – Reporte de anotaciones.....	15
Tabla 2.10.	Product Backlog.	17
Tabla 2.11.	Sprint 1 – Despliegue del entorno de desarrollo.	19
Tabla 2.12.	Sprint 2 – Extracción de políticas.	20
Tabla 2.13.	Sprint 3 – Reporte de anotaciones.....	20
Tabla 2.14.	Sprint 4 – Internacionalización.	21
Tabla 2.15.	Tecnologías usadas durante el desarrollo de la herramienta.....	28
Tabla 2.16.	Modificaciones del microservicio.	36
Tabla 2.17.	Versiones instaladas de ngx-translate.	50
Tabla 2.18.	Análisis de Aceptación HU-05, HU-06.	57
Tabla 2.19.	Análisis de Aceptación HU-07.....	62
Tabla 2.20.	Análisis de Aceptación HU-04.....	64
Tabla 2.21.	Resumen de Aceptación.	64

ÍNDICE DE CÓDIGOS

Código 2.1.	Configuración FTP y SSH.....	29
Código 2.2.	Script de Configuración Linux Ubuntu.....	30
Código 2.3.	Actualización de migraciones Tree-View.....	30
Código 2.4.	HTML Políticas.....	32
Código 2.5.	Consultar URLs Frontend.....	33
Código 2.6.	Contar URLs Backend.....	33
Código 2.7.	Conexión Backend y Servidor Linux.....	35
Código 2.8.	Modificación Ítem 1.....	36
Código 2.9.	Modificación Ítem 2.....	36
Código 2.10.	HTML Formulario Políticas.....	38
Código 2.11.	Clase PoliticaAnotadaReporte.....	40
Código 2.12.	Consulta SQL Reportes.....	40
Código 2.13.	Creación Archivo SCV.....	41
Código 2.14.	Creación de archivo YAML.....	43
Código 2.15.	HTML Reportes.....	44
Código 2.16.	Consultar Políticas Anotadas Frontend.....	45
Código 2.17.	Consultar Políticas Anotadas Backend.....	46
Código 2.18.	HTML Formulario Reportes.....	47
Código 2.19.	Validación Formulario Reportes.....	48
Código 2.20.	Envío de Políticas.....	48
Código 2.21.	HTML Header Principal.....	50
Código 2.22.	Añadir Idiomas.....	51
Código 2.23.	JSON de Traducciones.....	51
Código 2.24.	Traducciones en Archivos TS.....	52
Código 2.25.	Traducciones en Archivos HTML.....	52
Código 2.26.	Corrección Idioma de Módulos.....	52

RESUMEN

Para continuar con los esfuerzos por contribuir a la provisión de técnicas y herramientas para la evaluación de cumplimiento de requerimientos de privacidad y protección de datos, este trabajo presenta un conjunto de mejoras a la herramienta de anotación de políticas de privacidad, previamente presentada por Félix Miño [1]. Esta herramienta ya permitía la definición de diversos tipos de tratamientos de datos personales; no obstante, en este trabajo se ha desarrollado las siguientes mejoras: (i) un módulo para la carga automática de políticas de privacidad textuales a partir de una URL, (ii) un módulo para la generación de reportes sobre las anotaciones realizadas y (iii) se ha internacionalizado la aplicación.

Para ello, este trabajo ha sido organizado en tres capítulos. El capítulo 1 presenta el fundamento teórico sobre la privacidad, la protección de datos y el Proyecto de Ley Orgánica de Protección de Datos Personales (PLOPDP) ecuatoriano. Asimismo, se describe la metodología ágil Scrum y las tecnologías que fueron usadas para el desarrollo de la versión actualizada de la herramienta.

El capítulo 2 presenta los aspectos más relevantes del desarrollo de las nuevas funcionalidades de la herramienta. Primero, en el análisis, se usó historias de usuario y diagramas de caso de uso para la creación del *Product Backlog* y posteriormente los *Sprints Backlogs*, donde el objetivo principal es recolectar todos los requerimientos establecidos por el *Product Owner*. Segundo, en el diseño, se presenta los mockups de los *sprints* planteados. En la implementación, se ejecuta el conjunto de tareas y actividades de los *sprints* que suponen la realización del proyecto. En la validación y verificación, se ha efectuado un proceso de comprobación y análisis que aseguran que la herramienta que se ha desarrollado está acorde a las especificaciones del *Product Owner*.

El capítulo 3 presenta las conclusiones y recomendaciones obtenidas, una vez finalizado el trabajo. Se hace énfasis en la importancia y utilidad de la herramienta y en sugerencias para trabajos futuros.

Finalmente, se presenta las referencias bibliográficas y anexos del trabajo.

PALABRAS CLAVE: privacidad, datos personales, políticas de privacidad, Scrum, web, Angular, TypeScript, Flask

ABSTRACT

Aiming at contributing to the provision of techniques and tools for the compliance assessment with privacy and data protection requirements, this work presents a set of improvements to a privacy policy annotation tool, previously presented by Félix Miño [1]. This tool already allows the definition of different types of personal data practices; however, this work presents the following improvements: (i) a module for automated upload of textual privacy policies, (ii) a module for generating annotation reports and (iii) the application has been internationalized.

To this end, this work has been organized into three chapters. Chapter 1 presents the theoretical background on privacy, data protection and the Ley Orgánica de Protección de Datos Personales (LOPD). Furthermore, the agile Scrum methodology and the development technologies used in this work are described.

Chapter 2 presents the most relevant aspects of the development of the new functionalities of the tool. Firstly, in the analysis stage, the user stories and use case diagrams were specified to subsequently create the Product Backlog and, later, the Sprints Backlogs, where the main objective is to collect all the requirements established by the Product Owner. Secondly, in the design stage, the models of the proposed sprints are presented. In the implementation stage, the set of tasks and activities of the sprints that involve the completion of the project are executed. In the validation and verification stage, a testing and analysis process has been carried out to ensure that the developed tool complies with the Product Owner's specifications.

Chapter 3 presents the conclusions and recommendations. Emphasis is placed on the importance and usefulness of the tool and on suggestions for future work.

Finally, references and annexes of the work are presented.

KEY WORDS: privacy, personal data, privacy policies, Scrum, web, Angular, TypeScript, Flask

1 INTRODUCCIÓN

En estos días, es muy evidente el ritmo acelerado al que evoluciona la tecnología. Es muy palpable, por ejemplo, que el desarrollo de utilidades y servicios digitales nos simplifiquen tareas de la vida cotidiana y ofrecer mayores comodidades a los diferentes usuarios. No obstante, todo este avance ha facilitado que cualquier persona exponga sus datos (voluntaria e involuntariamente) y corra el riesgo de que se violen sus derechos de privacidad y protección de datos personales.

Este hecho ha llevado a que muchos países y estados regulen el tratamiento de datos personales por parte de las organizaciones. En Ecuador, se publicó en el Registro Oficial el 26 de mayo de 2021, la Ley Orgánica de Protección de Datos Personales (LOPDP), la cual tiene por objeto garantizar el derecho a la protección de datos personales, que incluye el acceso y decisión sobre la información y datos de este carácter [2].

La LOPDP estipula que las autoridades de control se deben encargar de que los sistemas de software cumplan el marco regulatorio. Siendo así, se requieren herramientas para la evaluación de aspectos de privacidad y protección de datos. Ciertos requisitos están relacionados con transparencia, en las cuales se requiere hacer uso de políticas de privacidad para informar a los usuarios como serán procesados sus datos personales. En este contexto, las autoridades de control requieren evaluar que las organizaciones cumplan los requisitos de transparencia. Para ello, se requiere extraer la práctica de privacidad que se encuentran definidas en las políticas textuales. No obstante, dada la cantidad de políticas de privacidad existentes dicha extracción no se lo puede llevar a cabo manualmente.

Es necesario entonces dirigir esfuerzos para crear herramientas que ayuden a extraer automáticamente las prácticas de privacidad. Como contribución, Félix Miño, estudiante de la Escuela Politécnica Nacional, desarrolló la herramienta web de anotación AnooTool-Cliip [1]. Aunque la herramienta web AnooTool-Cliip para anotación es muy útil, aún se requieren funcionalidades nuevas. Es así como en este trabajo se mejorará la herramienta para anotación de políticas de privacidad para (i) permitir la carga automática de políticas de privacidad textuales a partir de una URL, (ii) generar reportes sobre las anotaciones realizadas usando diferentes formatos y (iii) internacionalizar la aplicación.

1.1 OBJETIVOS

1.1.1 OBJETIVO GENERAL

- Implementar un conjunto de mejoras a la herramienta web AnooTool-Cliip, que permite realizar anotaciones de los tratamientos de datos personales en políticas de privacidad en español.

1.1.2 OBJETIVOS ESPECÍFICOS

- Estudiar los fundamentos teóricos referentes a la privacidad y protección de datos personales y de las tecnologías necesarias para el desarrollo de las nuevas funcionalidades requeridas.
- Estudiar la herramienta web AnooTool-Cliip.
- Especificar los requisitos del sistema con base en las nuevas funcionalidades requeridas.
- Diseñar los artefactos de software que soporten las nuevas funcionalidades requeridas.
- Implementar los artefactos diseñados.
- Validar y verificar que las nuevas funcionalidades se han desarrollado acorde a las especificaciones mencionadas.

1.2 ALCANCE

En el trabajo de Félix Miño se presentó el desarrollo de la herramienta web AnooTool-Cliip para anotación de prácticas de privacidad en políticas textuales. La herramienta fue desarrollada con lenguaje de programación TypeScript y el framework Angular. La Figura 1.1, muestra de manera resumida el funcionamiento de la herramienta de anotación.

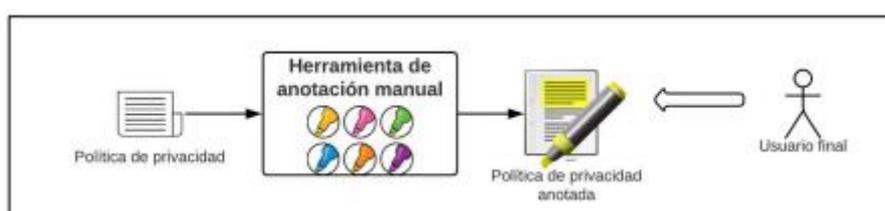


Figura 1.1. Funcionalidad de Herramienta de Anotación.

La herramienta requiere una política de privacidad textual como entrada. Un usuario, con rol de administrador, configura internamente las prácticas o tratamientos de protección de datos que se requiere anotar en una política. La herramienta divide una política de privacidad en segmentos o párrafos. Cada párrafo puede ser anotado por un usuario anotador, utilizando las prácticas o tratamientos previamente definidos. Finalmente, los resultados del proceso de anotación pueden ser visualizados por usuarios finales.

En el presente trabajo, primero se ha estudiado en detalle la versión anterior de la herramienta de anotación. Dado que se continuó con el desarrollo de una versión anterior de la herramienta, se tuvo primero que resolver las incompatibilidades debido a las actualizaciones de bibliotecas usadas o debido a actualizaciones de la plataforma subyacente. Posteriormente se desplegó y ejecutó la primera versión de la herramienta.

Al culminar este proyecto la herramienta permite (i) la carga automática de políticas de privacidad textuales a partir de una URL, (ii) la generalización de reportes sobre las anotaciones realizadas usando diferentes formatos y (iii) la internacionalización de la aplicación, soportando actualmente el inglés y español, aunque puede extenderse fácilmente a otros lenguajes.

El desarrollo de las nuevas funcionalidades de la herramienta web será guiado por la metodología de desarrollo ágil Scrum. Esta metodología permitirá abordar el proyecto de un modo iterativo e incremental. Además, el proceso de desarrollo está dividido en las 4 etapas mostradas en la Figura 1.2.

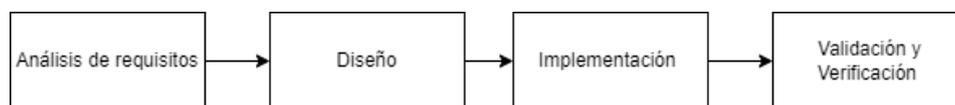


Figura 1.2. Método de desarrollo.

1.3 MARCO TEÓRICO

1.3.1 PRIVACIDAD Y PROTECCIÓN DE DATOS PERSONALES

PRIVACIDAD

Según el diccionario de Oxford Languages la privacidad se define como la parte más interior o profunda de la vida de una persona, que comprende sus sentimientos, vida familiar o relaciones de amistad [3]. Asimismo, el diccionario de la lengua española de la Real Academia Española define a la privacidad como el ámbito de la vida privada que se tiene derecho a proteger de cualquier intromisión [4].

En este sentido, la privacidad contempla diversos ámbitos de un individuo, incluyendo su comportamiento, sus rasgos físicos, sus intereses, sus comunicaciones, sus datos, etc., por lo que es un concepto abordado en el ámbito sociológico, cultural, legal y tecnológico. Debido a que muchos de los ámbitos antes descritos se desarrollan en el mundo digital, los riesgos a la privacidad de los usuarios se han hecho más evidentes. Siendo así, la privacidad se enfrenta a nuevos desafíos y plantea nuevas amenazas, en el contexto del desarrollo de la sociedad de la información, la expansión de la informática y de las telecomunicaciones [5].

DATOS PERSONALES Y SU PROTECCIÓN

En términos generales los datos personales son aquellos datos que nos permite identificar a una persona específica de un forma directa o indirecta. Los datos como nombre, foto,

número de teléfono o dirección permiten la identificación directa de un individuo. Asimismo, ciertos datos como la dirección de IP o nombre de usuario pueden permitir la identificación indirecta. Siendo así, en ambos casos estamos hablando de datos personales [6].

Los datos personales son parte de la privacidad de un individuo, por lo que dicho individuo tiene derecho a su protección. El término de protección de datos personales surge como un mecanismo para proteger pragmáticamente el derecho a la privacidad de las personas [7]. Carmela Troncoso [8] define tres categorías de tecnologías de protección a la privacidad de un individuo alineadas con los tres principales paradigmas de la privacidad reconocidos en el estado del arte. En la siguiente sección se describen estas tres categorías de tecnologías.

1.3.2 TECNOLOGÍAS PARA PROTECCIÓN DE LA PRIVACIDAD

PRIVACIDAD COMO CONFIDENCIALIDAD

Las tecnologías de esta categoría buscan con proteger la privacidad vista como el “derecho a estar solo”. Técnicamente, estas tecnologías buscan evitar que los personales de un individuo sean accesibles a cualquier entidad (en quien un individuo no confía). Así, el objetivo de estas tecnologías de privacidad es permitir el uso de servicios mientras se minimiza la cantidad de información expuesta. La información se refiere tanto a los datos intercambiados explícitamente con el servicio como a la información implícitamente disponible en los metadatos asociados con estos intercambios.

Para reducir la cantidad de información expuesta explícitamente intercambiada entre un usuario y un servicio x, actualmente hay dos tecnologías: cifrado y ofuscación. Las primeras aseguran que la información sea inteligible y que, por tanto, no se pueda inferir ningún tipo de información personal. Las segundas, a través de métodos de anonimización, generalización, supresión, adición ficticia, perturbación, entre otros, buscan que no sea posible asociar información a un individuo [8].

PRIVACIDAD COMO CONTROL

Las tecnologías de esta categoría asumen un paradigma de privacidad en que la divulgación de datos personales puede ser inevitable o es percibida como beneficiosa por un individuo. Técnicamente, estas tecnologías buscan que un individuo tenga control sobre los datos personales divulgados, por ejemplo, hacia una organización.

Estas tecnologías, por lo tanto, tienen dos objetivos principales: 1) permitir a los usuarios expresar cómo esperan que se utilicen los datos revelados a una organización, para entonces evitar el procesamiento no deseado de estos datos; y 2) permitir que las organizaciones definan y apliquen políticas que prevengan el uso indebido de la información, según lo definan los usuarios [8].

PRIVACIDAD COMO TRANSPARENCIA

Las tecnologías de esta categoría asumen un paradigma de privacidad basado en la transparencia sobre cómo los datos personales de un individuo son procesados. Estas tecnologías buscan que un individuo sea consciente sobre las implicaciones sus acciones.

Estas tecnologías (i) analizan las actividades en línea de un usuario para proporcionales retroalimentaciones acerca de sus acciones (p.ej., las implicaciones de permitir el acceso a cierta información personal en su terminal móvil), o (ii) auditan, a través de un tercero de confianza, registros de procesamiento de datos personales para verificar que no ha habido violaciones a la privacidad [8].

1.3.3 EVALUACIÓN DE CUMPLIMIENTO DE PRIVACIDAD Y PROTECCIÓN DE DATOS

En Ecuador, actualmente está en vigencia la Ley Orgánica de Protección de Datos Personales (LOPDP), que busca regular a las organizaciones en lo que respecta al tratamiento datos de carácter personal [2].

LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES (LOPDP)

El objetivo central de la LOPDP es mejorar la seguridad de la información en cuanto a los datos personales que manejan las diferentes organizaciones. Se pretende regular la confidencialidad de los datos personales y evitar que estos se usen con otros fines [2].

La LOPDP contiene 13 principios: juridicidad, lealtad, transparencia, finalidad (explícitas, legítimas y comunicadas la titular), pertenencia y minimización, proporcionalidad, confidencialidad, calidad y exactitud, conservación (durante un tiempo no mayor al necesario), seguridad, responsabilidad proactiva y demostrada, aplicación favorable al titular (Interpretación de la norma), independencia del control (autonomía e independencia del ente regulador). Estas son las bases de las medidas para evitar una manipulación inadecuada de los datos personales. El principio de lealtad dice que, si bien un tratamiento puede ser legítimo, no significa necesariamente que sea justo. El trato justo es una respuesta a las circunstancias, medios y resultados que el titular puede razonablemente esperar y que no le perjudican. Por tanto, debe incluir la identificación de todas las consecuencias del tratamiento sobre los derechos e intereses de las personas, titulares, grupos o colectivos interesados, con el fin de determinar si el impacto potencial es razonable.

El principio de transparencia se refiere a que la información o comunicación relativa al tratamiento de datos debe ser fácilmente accesible y fácil de entender. La transparencia está ligada a la lealtad, pues el titular, al conocer en un principio de las finalidades y condiciones del tratamiento, podrá dar un consentimiento informado. Precisamente, si este consentimiento es la base legitimadora del tratamiento de datos, conforme al art. 8 LOPDP, debe ser libre, específico, inequívoco e informado. Uno de los mecanismos de transparencia utilizados son las políticas de protección de datos.

POLÍTICAS DE PROTECCIÓN DE DATOS

Una política de protección de datos es un documento legal que plantea cómo una organización retiene, procesa o maneja los datos del usuario o cliente. La finalidad de las políticas de protección de datos es asegurar a los usuarios que sus datos serán recopilados para un fin específico y que cualquier actividad realizada será siempre con su consentimiento, protegiendo su privacidad. Según la LOPDP [2] en su artículo 47 donde se especifica las obligaciones del responsable y encargado del tratamiento de datos personales afirma que se tiene que implementar políticas de protección de datos personales afines al tratamiento de datos personales en cada caso en particular y se declara infracción en caso de no mantenerlas disponibles.

TRATAMIENTOS DE DATOS PERSONALES

Para los propósitos de este documento, un tratamiento de datos personales es el ejercicio o realización de una actividad sobre los datos de carácter personal, de forma continuada y conforme a las reglas estipuladas en una política de privacidad [9]. Algunos ejemplos de prácticas de privacidad son la recolección, transferencia y almacenamiento de datos personales.

1.3.4 ANNOTOOL-CLIP - HERRAMIENTA PARA LA ANOTACIÓN DE TRATAMIENTOS DE DATOS PERSONALES

AnooTool-Cliip es una herramienta web para la anotación de tratamientos de datos expresados en políticas de privacidad en español [1]. La herramienta fue desarrollada con el lenguaje de programación TypeScript y el framework Angular.

La Figura 1.3 muestra la arquitectura de la herramienta, la cual se divide en dos componentes web:

- 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, que fue desarrollado con el lenguaje de programación Python y el framework Flask. En resumen, la herramienta web posee tres roles (administrador, anotador y usuario final), dos tipos de sitios web (público y privado) y cuatro módulos principales (administración, anotación, consolidación y visualización).

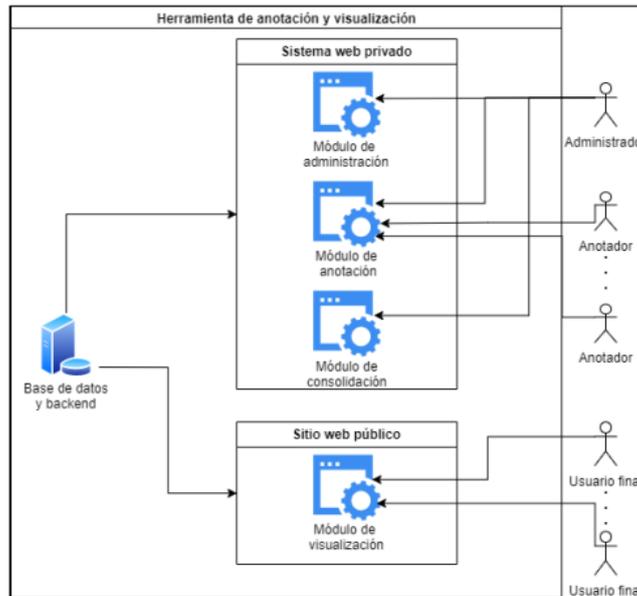


Figura 1.3. Despliegue de la Herramienta de Anotación.

La Tabla 1.1 muestra los roles y las funcionalidades asociadas a cada uno.

Tabla 1.1. Roles y Módulos.

			Roles		
			Rol Administrador	Rol Anotador	Usuario final
Sitios web	Privado (requiere autenticación)	Módulo Administración	✓	✗	✗
		Modulo Anotación	✓	✓	✗
		Modulo Consolidación	✓	✗	✗
	Público (no requiere autenticación)	Modulo Visualización	✓	✓	✓

La herramienta es capaz de soportar acceso concurrente de múltiples usuarios autenticados o no autenticados, de manera que los usuarios con rol anotador o rol administrador puedan trabajar sobre la misma política de privacidad y los usuarios finales puedan observar las anotaciones realizadas.

Se presenta una breve descripción sobre el funcionamiento de cada módulo:

- **Módulo de administración.** – Este módulo contiene los siguientes submódulos:

- a. Submódulo de administración de usuarios. – Permite realizar las operaciones CRUD¹ de usuarios.
 - b. Submódulo de administración de políticas. – Permite realizar las operaciones CRUD de las políticas de privacidad.
 - c. Submódulo de administración de tratamientos. – Permite realizar las operaciones CRUD de los tratamientos de datos personales.
 - d. Submódulo de administración de atributos. – Permite realizar las operaciones CRUD de los atributos de un tratamiento de datos personales.
 - e. Submódulos de administración de valores. – Permite realizar las operaciones CRUD de los valores que podría tener un atributo.
- Módulo de anotación. – Permite a los usuarios con rol anotador o administrador realizar y editar anotaciones de tratamiento de datos sobre políticas de privacidad, que les fueron previamente asignadas. Solo se puede editar anotaciones realizadas por el mismo usuario, en ningún caso se permite anotación realizadas por otros usuarios.
 - Módulo de consolidación. – Permite al usuario con rol administrador consolidar una política de privacidad previamente anotada, es decir, fijar una anotación final que represente al conjunto de usuarios.
 - Módulo de visualización. – El objetivo de este módulo es que cualquier usuario ingrese y visualice las políticas de privacidad anotadas juntamente con los comentarios [1].

1.3.5 HERRAMIENTAS DE DESARROLLO UTILIZADAS EN EL TRABAJO

ANGULAR

Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página (SPA-Single Page Application). Angular separa completamente el *frontend* y el *backend* en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez. Además, posibilita modificaciones y actualizaciones [10].

PYTHON

Python es un lenguaje de programación de alto nivel y orientado a objetos. Es muy atractivo en el campo del Desarrollo Rápido de Aplicaciones (RAD) porque ofrece tipificación dinámica y opciones de encuadernación dinámicas. Python es relativamente simple de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes [11].

¹ **CRUD:** agrupación de cuatro acciones: Create (crear), Read (leer), Update (actualizar) y Delete (eliminar), operaciones básicas que se puede realizar sobre un conjunto de datos

FLASK

Flask es un “micro” Framework escrito en Python que fue concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC. La palabra “micro” no designa a que sea un proyecto pequeño o que nos permita hacer páginas web pequeñas, sino que al instalar Flask tenemos las herramientas necesarias para crear una aplicación web funcional. No obstante, si se necesita una nueva funcionalidad, hay un conjunto muy grande extensiones (*plugins*) que se pueden instalar con Flask que le van dotando de funcionalidad adicional [12].

1.3.6 METODOLOGÍA ÁGIL SCRUM

Scrum es en realidad un marco de trabajo que define tres roles, cinco tipos de eventos y tres artefactos. La Figura 1.4 muestra los elementos de este marco de trabajo, los mismos que son descritos en las siguientes secciones tomando como referencia la guía Scrum de Ken Schwaber y Jeff Sutherland [13].

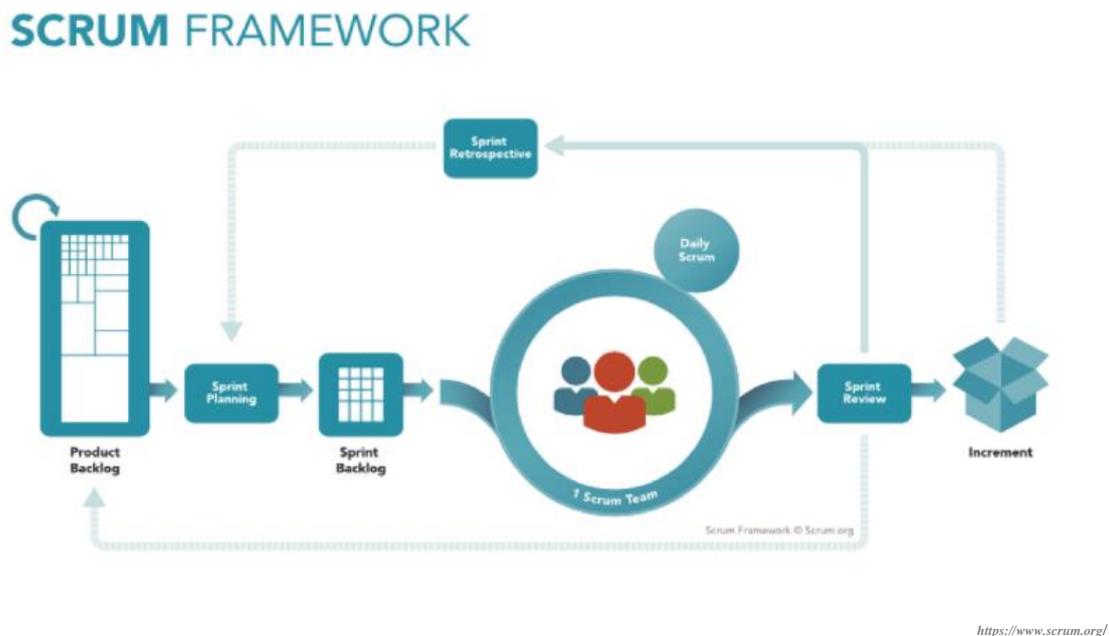


Figura 1.4. Marco de trabajo Scrum [14].

SCRUM TEAM O ROLES SCRUM

La unidad básica de Scrum es un pequeño grupo de personas denominadas *Scrum Team*. El *Scrum Team* está conformado por el *Scrum Máster*, el *Product Owner* y los *Desarrolladores*. No hay divisiones, sub-equipos, o jerarquía en un *Scrum Team*, ya que se trata de un equipo cohesionado de profesionales que se centran en un objetivo al mismo tiempo.

Los *Scrums Teams* son multifuncionales, por lo que sus miembros tienen todas las habilidades necesarias para crear valor en cada *sprint*. También son autónomos, por lo que deciden internamente quién hace qué, cuándo y cómo. Los equipos de Scrum son lo

suficientemente pequeños para ser ágiles y lo suficientemente grandes para realizar un trabajo importante en un *sprint*. Generalmente, un equipo no supera las diez personas.

- **Product Owner.** - El *Product Owner* es quien posee comunicación directa con los clientes. El Product Owner es responsable de maximizar el valor del producto creado por el trabajo del equipo Scrum. También es responsable de la gestión eficaz del Product Backlog, a través de las siguientes actividades:
 - Desarrollar y comunicar claramente los objetivos del producto;
 - Crear y comunicar claramente los elementos del Product Backlog;
 - Ordenar elementos del Product Backlog; y,
 - Asegurarse de que el Product Backlog sea transparente, visible y fácil de entender.
- **Scrum Master.** - El *Scrum Master* es responsable de seguir Scrum como se define en el manual de Scrum. Lo hace ayudando a todos a comprender la teoría y la práctica de Scrum en los equipos y organizaciones. El Scrum Master es responsable de la efectividad del *Scrum Team*.
- **Developers.** – Las personas profesionales del *Scrum Team* que trabajan en todos los aspectos para crear un paso práctico para cada *sprint* son los *Developers* o desarrolladores. Las habilidades específicas que necesitan los desarrolladores suelen ser amplias y varían según el campo de trabajo.

EVENTOS

- **El sprint.** - El *sprint* es un ciclo o iteración en el que se completa una cantidad de trabajo establecida, Estos son eventos de duración fija que duran un mes o menos. Un nuevo *sprint* comienza inmediatamente después de que finaliza el *sprint* anterior.
- **Sprint planning.** - El *Sprint Planning* es una reunión del *Scrum Team* donde el equipo se organiza y eligen las diferentes tareas a realizar durante todo el Sprint. El *Scrum Team* crea este plan resultante mediante trabajo colaborativo.
- **Daily scrum.** - *Daily Scrum* son reuniones diarias de 15 minutos para los desarrolladores del *Scrum Team*. El propósito de la *Daily Scrum* es inspeccionar el progreso hacia el objetivo del *sprint* y adaptar el *Sprint Backlog* según sea necesario, ajustando para ello el trabajo planificado entrante. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del *sprint*.
- **Sprint review.** - El propósito de *Sprint Review* es verificar los resultados del sprint e identificar ajustes futuros. Los equipos Scrum presentan los resultados de su trabajo a las partes interesadas clave y analizan el progreso hacia el objetivo del producto.
- **Sprint retrospective.** -El propósito de la *Sprint Retrospective* es planificar formas de aumentar la calidad y la efectividad. Se identifican los supuestos que los llevaron por mal camino y se exploran sus orígenes. El *Scrum Team* analiza qué salió bien durante el *sprint*, qué problemas encontró y cómo se resolvieron (o no) esos problemas.

ARTEFACTOS DE SCRUM

- **Product backlog.** - El *Product Backlog* es una lista priorizada y organizada de lo que se debe hacer para llevar a cabo el producto. En el Product Backlog se tratan temas netamente funcionales, mas no temas técnicos y es el *Product Owner* quien está al mando de este artefacto. El refinamiento del *Product Backlog* es el proceso de desglosar y definir aún más los items del *Product Backlog* en unidades más pequeñas y precisas.
- **Sprint backlog.** -EL *Sprint Backlog* es un subconjunto del *Product backlog* y consta del objetivo del *sprint* (por qué), un conjunto de elementos del *Product Backlog* seleccionados para el *sprint* (qué) y un plan de acción para el crecimiento (cómo). El *Sprint Backlog* es un plan diseñado por y para desarrolladores.
- **Incrementos.** - Los *incrementos* son pasos específicos para lograr los objetivos del producto. Cada paso se basa en todos los pasos anteriores y se verifica cuidadosamente para garantizar que todos los pasos funcionen juntos. El crecimiento debe estar disponible para proporcionar valor [14].

2 METODOLOGIA

En este capítulo se presenta la metodología utilizada y los aspectos relevantes del desarrollo de las nuevas funcionalidades de la herramienta de anotación. En la primera sección, se presenta el análisis, donde se usó historias de usuario para la creación del *Product Backlog* y posteriormente los *Sprints Backlogs*. En la segunda sección se presenta, el diseño de los *sprints* planteados. En la tercera sección se presenta la implementación de los *sprints*. Finalmente, en la cuarta sección se presenta la validación y verificación asegurando que la herramienta este acorde a las especificaciones del *Product Owner*. Se utilizó la metodología de desarrollo ágil Scrum, que fue descrita en la sección 1.3.6.

En la Tabla 2.1 se presenta los roles Scrum definidos para el proyecto:

Tabla 2.1. Roles Scrum.

Rol	Nombre
Scrum Máster	Danny Guamán
Product Owner	Danny Guamán
Developer	David Proaño

2.1 ANÁLISIS

Esta sección abarca desde la creación del *Product Backlog*, hasta la creación de los *Sprints Backlogs*, junto con todos los subprocesos que demanda la guía Scrum. El objetivo principal es recolectar todos los requerimientos establecidos por el *Product Owner* basados en las necesidades de los usuarios que manejarán la herramienta de anotación.

2.1.1 REQUISITOS NO FUNCIONALES

Los requisitos no funcionales requeridos por el *Product Owner* son:

- Usar las mismas tecnologías con las que se desarrolló la primera versión de la herramienta para desarrollar las nuevas funcionalidades
- Resolver las incompatibilidades debido a actualizaciones de bibliotecas usadas en versiones anteriores o debido a actualizaciones de la plataforma subyacente.
- Internalizar la herramienta para soportar al menos 2 idiomas: inglés y español.

2.1.2 REQUISITOS FUNCIONALES

Los requisitos funcionales requeridos por el *Product Owner* son:

- Descargar automáticamente el texto de una política de privacidad, a partir de su URL.
- Descargar automáticamente los textos de conjunto de políticas de privacidad, partir de sus URLs.
- Generar reportes en un formato semi-estructurado especificado por el usuario, incluyendo CSV y YAML

2.1.3 HISTORIAS DE USUARIO

Sobre la base de los requisitos iniciales definidos en la sección anterior, se definieron las historias de usuario usando el formato mostrado en la Tabla 2.2.

Tabla 2.2. Formato de Historia de Usuario.

Código	HU-00
Nombre	Nombre descriptivo
Descripción	Yo [<i>nombre solicitador</i>] como [<i>rol de la herramienta de anotación</i>] deseo [<i>funcionalidad</i>] para [<i>objetivo</i>]
Criterios de aceptación	<ul style="list-style-type: none">• Debe cumplir [<i>condición 1</i>]• Debe cumplir [<i>condición 2</i>]• ...• Debe cumplir [<i>condición n</i>]

En la Tabla 2.3 se muestra la historia de usuario para la adaptación de tecnologías de desarrollo.

Tabla 2.3. Historia de Usuario HU-01 – Adaptación de tecnologías.

Código	HU-01
Nombre	Adaptación de tecnologías

Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que todo el conjunto de nuevas funcionalidades sea implementado con las mismas tecnologías con las que se desarrolló la primera versión de la herramienta para que exista compatibilidad de la aplicación y facilidad de desarrollo.
Criterios de aceptación	Debe cumplir que la PC de desarrollo tenga las mismas tecnologías usadas en la primera versión de la herramienta.

En la Tabla 2.4 se muestra la historia de usuario para la resolución de problemas de compatibilidad debido a actualizaciones.

Tabla 2.4. Historia de Usuario HU-02 – Compatibilidad de actualizaciones.

Código	HU-02
Nombre	Compatibilidad de actualizaciones
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que se resuelva las incompatibilidades debido a actualizaciones de bibliotecas usadas en versiones anteriores o debido a actualizaciones de la plataforma subyacente para evitar que las diferentes tecnologías de desarrollo, bibliotecas o paquetes presentes problemas de incongruencia en las nuevas versiones.
Criterios de aceptación	Debe cumplir que haya cero problemas de incongruencia entre las versiones de tecnologías, bibliotecas y/o paquetes a usar

En la Tabla 2.5 se muestra la historia de usuario para el despliegue de la versión original de la herramienta de anotación.

Tabla 2.5. Historia de Usuario HU-03 – Despliegue de herramienta.

Código	HU-03
Nombre	Despliegue de la herramienta de anotación
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta de anotación sea estudiada, desplegada y ejecutada, para que los desarrolladores estén familiarizados con todas las funcionalidades implementadas en la primera versión.
Criterios de aceptación	Debe cumplir que la herramienta sea correctamente manipulada en todos sus módulos.

En la Tabla 2.6 se muestra la historia de usuario para la internacionalización de la herramienta.

Tabla 2.6. Historia de Usuario HU-04 – Internacionalización de la aplicación.

Código	HU-04
Nombre	Internacionalización de la aplicación
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo adaptar a la herramienta para mostrar información en al menos dos idiomas: inglés y español, para presentar a usuarios que hablan diferentes idiomas, una aplicación visual y funcionalmente idéntica.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que la herramienta se despliegue inicialmente en idioma inglés. • Debe cumplir que el usuario este en la capacidad de elegir el idioma en que la herramienta despliegue todos sus módulos en cualquier momento.

En la Tabla 2.7 se muestra la historia de usuario para la extracción automática de políticas de privacidad de forma individual.

Tabla 2.7. Historia de Usuario HU-05 – Extracción de políticas individuales.

Código	HU-05
Nombre	Extracción automática de políticas individuales
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta sea capaz de extraer automáticamente el texto de una política de privacidad, mediante una URL, para facilitar este proceso a los usuarios.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que el usuario ingresa una URL. • Debe cumplir que la herramienta extrae el texto de la política de privacidad automáticamente. • Debe cumplir que el usuario ingrese nombre y fecha de la política. • Debe cumplir que el usuario pueda editar el texto extraído. • Debe cumplir que se pueda guardar la política.

En la Tabla 2.8 se muestra la historia de usuario para la extracción automática de políticas de privacidad de forma grupal.

Tabla 2.8. Historia de Usuario HU-06 – Extracción de políticas grupales.

Código	HU-06
Nombre	Extracción automática de políticas grupales
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta extraiga automáticamente el texto de un conjunto de políticas de privacidad mediante la asignación de una ruta con dirección a un fichero donde están alojadas varias URLs, para optimizar tiempo al ingresar URLs una por una y facilitar el proceso a los usuarios.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que la herramienta pida una ruta de un fichero. • Debe cumplir que se extraiga el texto de un conjunto de políticas de privacidad automáticamente a la vez. • Debe cumplir que el usuario ingrese nombre y fecha de las políticas. • Debe cumplir que el usuario pueda editar los textos extraídos. • Debe cumplir que se pueda guardar las políticas.

En la Tabla 2.9 se muestra la historia de usuario para la generación de reportes de anotaciones.

Tabla 2.9. Historia de Usuario HU-07 – Reporte de anotaciones.

Código	HU-07
Nombre	Reporte de anotaciones
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta sea capaz de crear reportes en un formato semiestructurado especificado por el usuario, para crear datasets.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que el usuario elija si se muestra políticas anotadas finalizadas o no finalizadas. • Debe cumplir que la herramienta muestre una tabla con políticas anotadas. • Debe cumplir que el usuario pueda escoger una o muchas políticas anotadas. • Debe cumplir que el usuario pueda escoger un formato para el reporte.

- Debe cumplir que la herramienta genere el reporte de las políticas anotadas en el formato específico.

2.1.4 DIAGRAMA DE CASOS DE USO

En la Figura 2.1 se muestra el diagrama de casos de uso que incluye las nuevas funcionalidades que deberá soportar la herramienta.

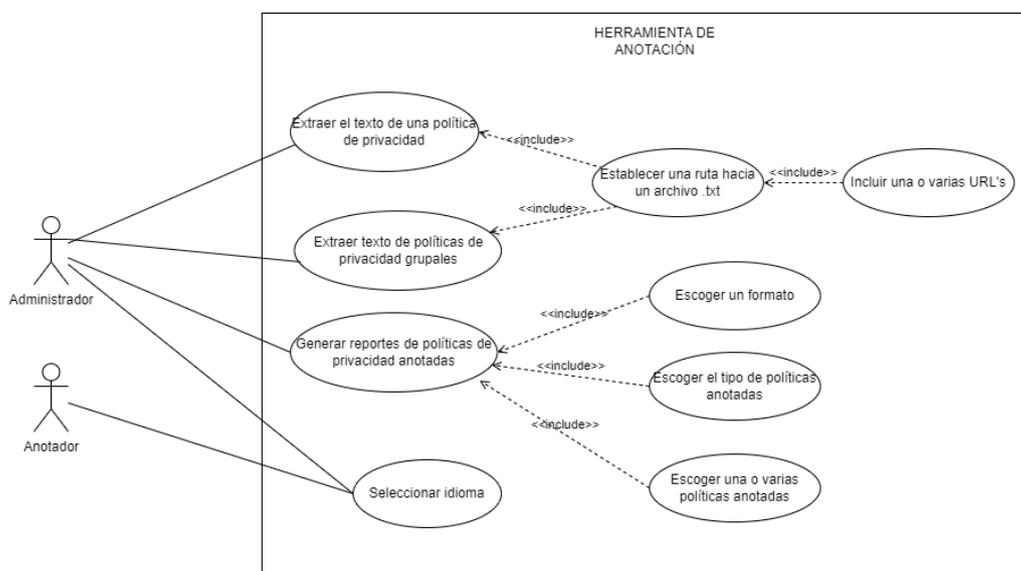


Figura 2.1. Diagrama de casos de uso de la herramienta de anotación.

- Un usuario con rol Administrador tendrá un módulo para extraer el texto de una política de privacidad automáticamente al establecer una ruta hacia un fichero que contiene una URL.
- Un usuario con rol Administrador tendrá un módulo para extraer el texto de varias políticas de privacidad automáticamente al establecer una ruta hacia un fichero que contiene varias URLs.
- Un usuario con rol Administrador tendrá a su disposición un módulo para generar reportes de políticas que hayan sido previamente anotadas, el usuario debe escoger las políticas y establecer un formato.
- Un usuario con rol Administrador o Anotador podrá escoger el idioma, entre inglés y español, en que la herramienta despliegue todos sus módulos.

2.1.5 PRODUCT BACKLOG

Una vez establecidas las historias de usuario se ha creado el *Product Backlog*. Para ello, se ha llevado a cabo un proceso de refinamiento, que consiste en desglosar y definir las historias de usuario en unidades más pequeñas y precisas denominadas tareas. De esta manera, se ha logrado tener detalle, orden y prioridad.

Debido a que Scrum es un marco de trabajo iterativo y muy susceptible a cambios, el *Product Backlog* se ha actualizado constantemente, siempre bajo supervisión del *Product Owner*.

En la Tabla 2.10 se presenta el *Product Backlog* final organizado por prioridad.

Tabla 2.10. Product Backlog.

Código	Nombre historia de usuario	Código tarea	Nombre tarea	Descripción	Prioridad
HU-01	Adaptación de tecnologías	T-01	Adaptación de tecnologías	Instalación de las mismas tecnologías usadas para el desarrollo de la primera versión de la herramienta: Python, Flask, Angular.	Alta
HU-02	Compatibilidad de actualizaciones	T-02	Compatibilidad de actualizaciones	Instalación de paquetes y bibliotecas necesarias y resolución de incompatibilidades debido a actualizaciones.	Alta
HU-03	Despliegue de la herramienta de anotación	T-03	Despliegue de la herramienta de anotación	Despliegue y manipulación funcional de la primera versión de la herramienta.	Alta
HU-05 / HU-06	Políticas individuales / Políticas grupales	T-04	Servidor Ubuntu	Creación de un servidor Ubuntu	Alta
		T-05	Despliegue microservicio	Despliegue del microservicio 3 del trabajo “Desarrollo de microservicios para la extracción de políticas de privacidad desde artefactos de una aplicación móvil android” desarrollado por Julio Cesar Miranda, en el servidor Ubuntu.	Alta
		T-06	Automatización Servidor Linux	Creación de un Script de Linux para automatizar configuración del servidor y	Alta

				despliegue del microservicio.	
		T-07	Tree-view	Actualización del menú principal de la herramienta de manera que se muestren los nuevos módulos a incorporar.	Media
		T-08	Frontend políticas	Creación de interfaz gráfica (Frontend) para insertar políticas automáticamente.	Media
		T-09	Lógica Frontend de políticas	Creación de las funciones necesarios en el frontend para que el usuario pueda insertar una ruta con dirección a un archivo en formato TXT que contiene un conjunto de URLs y enviarlo al backend. (requerimientos del microservicio)	Media
		T-10	Conexión Backend y Servidor Linux	Creación de Script en Python para conexión entre backend de la herramienta de anotación y el nuevo servidor Ubuntu y envió del archivo en formato TXT.	Media
		T-11	Modificación microservicio	Adaptación y modificación del microservicio hacia las nuevas necesidades.	Media
		T-12	Validación formulario políticas	Generación de formulario en el frontend para insertar datos de la política o modificar el texto de esta. Creación de lógica necesaria en el backend para guardar la nueva	Media

				política en la base de datos.	
HU-07	Reportes	T-13	Consulta SQL	Creación de consulta SQL usando SQLAlchemy, con los datos necesarios para realizar reportes de políticas anotadas.	Media
		T-14	Reportes en CSV	Creación de Script de Python que genere reportes de políticas anotadas en formato CSV.	Media
		T-15	Reportes en YAML	Creación de Script de Python que genere reportes de políticas anotadas en formato YAML.	Media
		T-16	Tabla de anotaciones	Generación de interfaz gráfica que muestre una tabla con las políticas anotadas.	Media
		T-17	Validación formulario reportes	Creación y validación de formulario para escoger el tipo de políticas, el formato del reporte y las políticas anotadas.	Media
HU-04	Internacionalización de la aplicación	T-18	Internacionalización de la aplicación	Creación de funciones necesarias para la traducción de textos entre inglés y español.	Baja

2.1.6 SPRINT BACKLOG

El *Sprint Backlog* esta más susceptible a cambios en los eventos Srum a lo largo del desarrollo del trabajo, incluso más que el *Product Backlog*.

En la Tabla 2.11 se detalla el *sprint* 1 junto con sus tareas a realizar, donde el objetivo es ejecutar y estudiar la primera versión de la herramienta.

Tabla 2.11. Sprint 1 – Despliegue del entorno de desarrollo.

N° Sprint	1
Nombre Sprint	Despliegue del entorno de desarrollo

Objetivo Sprint		Ejecutar y estudiar la primera versión de la herramienta de anotación		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-01	Adaptación de tecnologías	T-01	Alta	1
HU-02	Compatibilidad de actualizaciones	T-02	Alta	2
HU-03	Despliegue de la herramienta de anotación	T-03	Alta	3

En la Tabla 2.12 se detalla el *sprint* 2 junto con sus tareas a realizar, donde el objetivo es crear un nuevo módulo para la extracción automática de políticas individuales y grupales mediante URLs.

Tabla 2.12. Sprint 2 – Extracción de políticas.

N° Sprint		2		
Nombre Sprint		Extracción de políticas		
Objetivo Sprint		Crear módulo de extracción automática de políticas individuales y grupales		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-05 / HU-06	Políticas Individuales / Políticas Grupales	T-04	Alta	1
		T-05	Alta	3
		T-06	Media	2
		T-07	Media	3
		T-08	Media	2
		T-09	Media	3
		T-10	Media	5
		T-11	Media	1
T-12	Media	5		

En la Tabla 2.13 se detalla el *sprint* 3 junto con sus tareas a realizar, donde el objetivo es crear un nuevo módulo para la generación de reportes de políticas anotadas.

Tabla 2.13. Sprint 3 – Reporte de anotaciones.

N° Sprint		3		
Nombre Sprint		Reporte de anotaciones		
Objetivo Sprint		Crear módulo de reportes		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-07	Reportes	T-13	Media	5

		T-14	Media	3
		T-15	Media	3
		T-16	Media	3
		T-17	Media	3

En la Tabla 2.14 se detalla el *sprint* 2 junto con sus tareas a realizar, donde el objetivo es internacionalizar a la herramienta de anotación.

Tabla 2.14. Sprint 4 – Internacionalización.

N° Sprint		4		
Nombre Sprint		Internacionalización		
Objetivo Sprint		Internacionalizar la herramienta de anotación		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-04	Internacionalización de la aplicación	T-18	Baja	5

2.2 DISEÑO

En esta sección se presenta el diseño general de los cuatro *sprints* planteado en la sección anterior.

2.2.1 DISEÑO SPRINT 1 – DESPLIEGUE DEL ENTORNO DE DESARROLLO

El *sprint* 1 no contempla el diseño de ningún tipo de artefacto, ya que el objetivo de este *sprint* fue desplegar el ambiente de desarrollo.

2.2.2 DISEÑO SPRINT 2 - EXTRACCIÓN DE POLITICAS

La extracción de políticas se refiere a extraer el texto de una política de privacidad automáticamente mediante la asignación de una URL. Para el diseño de este *sprint* se ha incluido al microservicio² de Julio Cesar Miranda [15] que extrae y almacena el texto y código HTML de una política de privacidad de diversas fuentes, incluyendo páginas HTML que utilizan diversas tecnologías web, documentos PDF y documentos web.

Al incluir al microservicio aumenta un nuevo componente, por lo cual, debe haber un rediseño en la arquitectura general de la herramienta. A continuación, se presenta el diseño del *sprint* 2 que consta del rediseño de la arquitectura de la herramienta, diseño de diagramas de actividades y diseño de interfaces.

² **Microservicio:** en el desarrollo de software, son sistemas que permiten desarrollar aplicaciones que se componen de módulos independientes [19]

REDISEÑO DE ARQUITECTURA

El microservicio debe ser desplegado en un servidor Linux Ubuntu 18, ya que así lo recomienda el autor, con el fin de no tener (en la medida de lo posible) problemas de compatibilidad. La Figura 2.2 muestra la arquitectura rediseñada, que consta de los siguientes componentes: [15].

- Dos aplicaciones web: herramienta de anotación y herramienta de visualización, las cuales conforman el *frontend* de la herramienta.
- Una aplicación de servidor que se comunica con la base de datos y ejecuta las acciones solicitadas por las aplicaciones web, la cual conforma el *backend* de la herramienta.
- Un servidor Linux Ubuntu 18, donde está alojado el microservicio para la extracción de textos de políticas de privacidad.

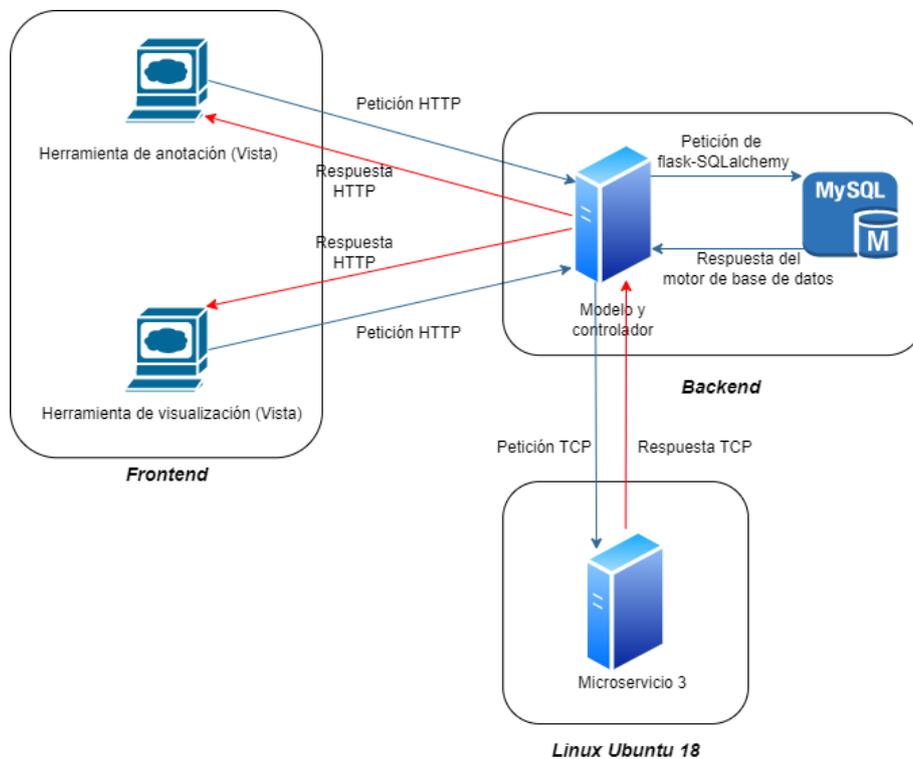


Figura 2.2. Arquitectura rediseñada de la herramienta.

DISEÑO DE ACTIVIDADES

La Figura 2.3 presenta el diseño del diagrama de actividades, que permite visualizar el funcionamiento requerido en este *sprint*.

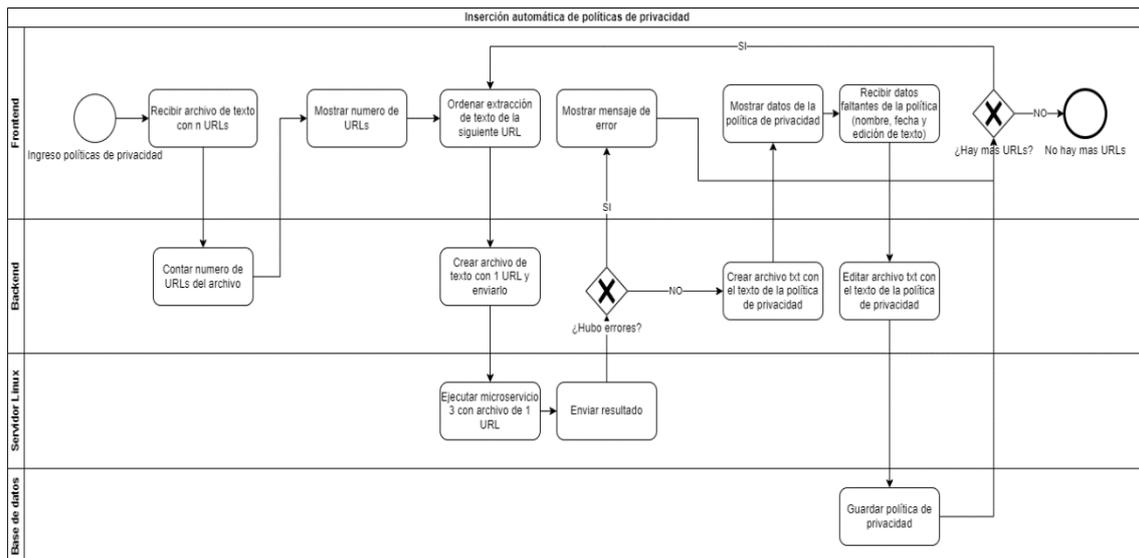


Figura 2.3. Diagrama de Actividades Políticas.

En este *sprint* hay comunicación entre las cuatro partes de la aplicación: *frontend*, *backend*, servidor Linux y base de datos. El *frontend* envía un archivo de texto que contiene un conjunto de URLs de políticas de privacidad hacia el *backend*. El *backend* verifica el archivo de texto con las URLs y las envía hacia el servidor Linux. El servidor Linux ejecuta el microservicio con el archivo de URLs como input y devuelve el texto extraído nuevamente al *backend*. El *backend* envía la respuesta al *frontend* y este muestra el resultado al usuario. El usuario puede editar el texto descargado e insertar los datos de la política de privacidad que son nombre y fecha y envía una vez más al *backend* para este finalmente enviar a la base de datos para guardar la política.

DISEÑO DE INTERFACES

Para el diseño de las interfaces se ha usado la herramienta *Figma*³, que permite la generación en línea de mockups.

En este *sprint* está incluida la tarea de modificación del *tree-view*⁴ principal de la herramienta. En la Figura 2.4 se presenta el rediseño del *tree-view*, donde se puede observar en la parte izquierda de la figura que contiene nuevos módulos que son: **Políticas** con los hijos **Gestión** e **Insertar** estos permiten la extracción de los textos de las políticas de privacidad, y el módulo **Reportes** para la generación de reportes de políticas de privacidad anotadas.

³ **Figma:** editor de gráficos vectorial y herramienta de generación de prototipos, basada en la web [20].

⁴ **Tree-view:** son elementos de interfaz gráfica de usuario que presenta una vista jerárquica de información.

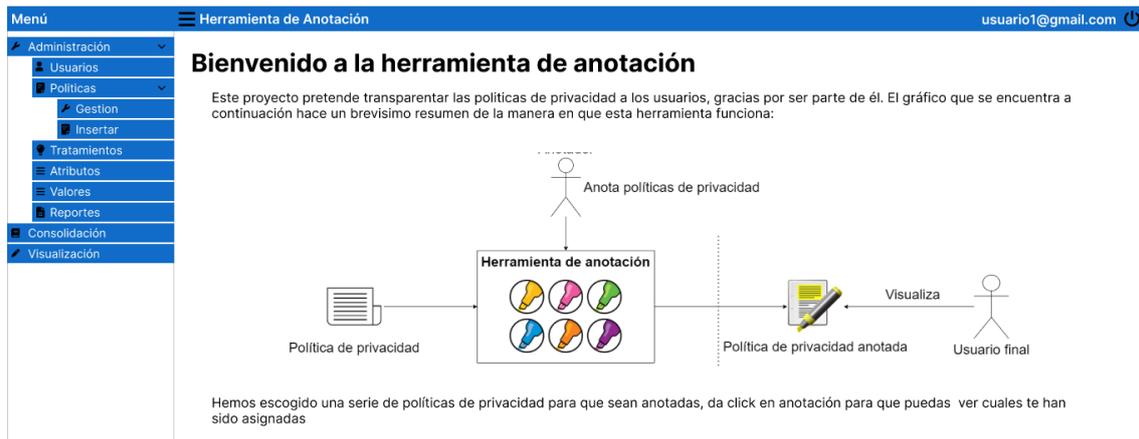


Figura 2.4. Diseño Interfaz Bienvenida.

En la Figura 2.5 se presenta el diseño de una ventana informativa, la cual contiene los pasos que se debe seguir para insertar una o varias políticas de privacidad de forma correcta. El usuario con rol Administrador podrá acceder a esta ventana en el módulo *Insertar* en la sección de *Políticas* al dar *click* en el icono *Ayuda* de la parte superior derecha.



Figura 2.5. Diseño Ayuda Extracción de Texto de Política de Privacidad.

En la Figura 2.6 se presenta el diseño de la interfaz del módulo *Administración* → *Políticas* → *Insertar* donde se podrá insertar el fichero con un conjunto de URLs, visualizar el resultado, llenar los datos de la política de privacidad y finalmente guardarla en la base de datos de la aplicación web.

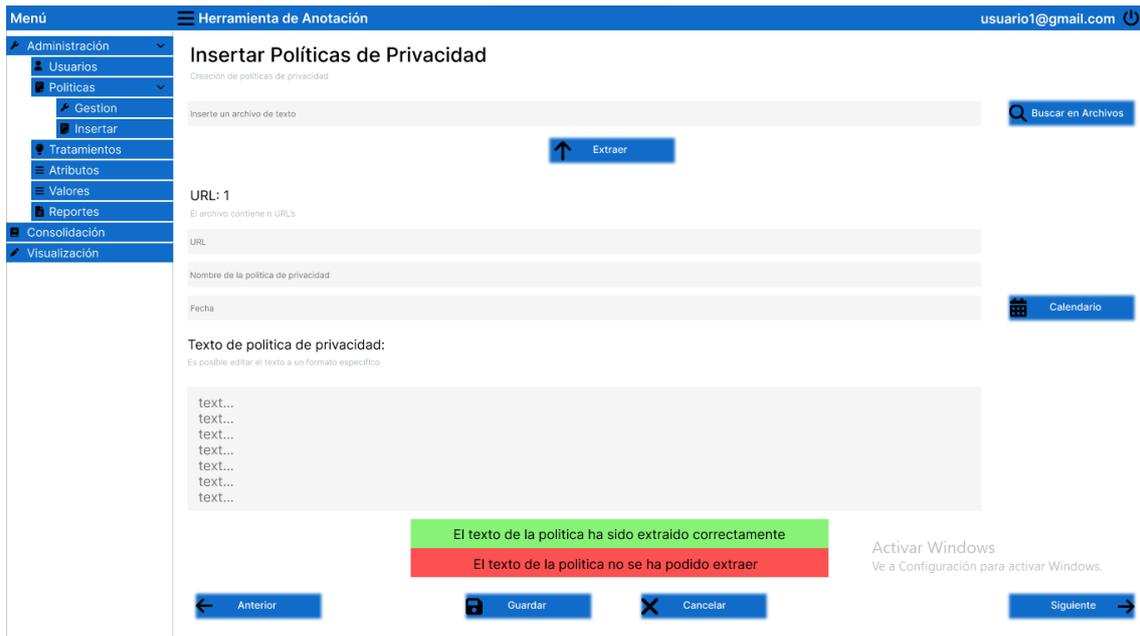


Figura 2.6. Diseño Interfaz Políticas.

2.2.3 DISEÑO SPRINT 3 - GENERACIÓN DE REPORTES

El objetivo de este *sprint* es generar un nuevo módulo llamado Reportes, para generar *datasets*⁵ de anotaciones de diferentes políticas de privacidad. Se presenta el diseño general del *sprint* 3 que consiste en el diseño del diagrama de actividades para la generación de reportes y diseño de interfaces.

DISEÑO DE DIAGRAMA DE ACTIVIDADES

La Figura 2.7 presenta el diseño de diagramas de actividades para visualizar el funcionamiento del *sprint*, que consiste en la generación de un módulo de reportes.

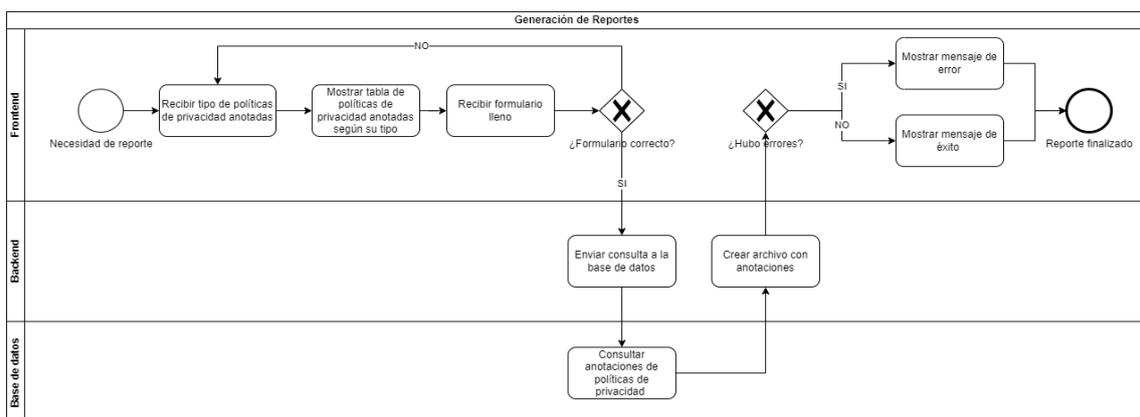


Figura 2.7. Diagrama de Actividades Reportes.

⁵ **Datasets:** conjunto de datos habitualmente tabulados.

En este *sprint* hay comunicación entre las tres partes de la aplicación: *frontend*, *backend* y base de datos. El *frontend* espera recibir el tipo de políticas anotadas. Estas se dividen en dos grupos: *Finalizadas* y *No Finalizadas*. Una vez que el usuario elige una opción el *frontend* envía los datos. El *backend* realiza una consulta a la base de datos y devuelve toda la información de las anotaciones de las políticas de privacidad finalizadas o no finalizadas según la preferencia del usuario. El *frontend* ordena la información y la muestra en forma de tabla al usuario. El usuario elige las anotaciones y el formato del *dataset*. El *frontend* valida todas las opciones escogidas y envía nuevamente al *backend*. Finalmente, el *backend* realiza otra consulta a la base de datos y crea el reporte según las especificaciones recibidas.

DISEÑO DE INTERFACES

En la Figura 2.8 se presenta el diseño de la interfaz del módulo *Administración* → *Reportes* donde se podrá elegir las anotaciones de políticas de privacidad y generar uno o varios reportes en base a un formato específico.

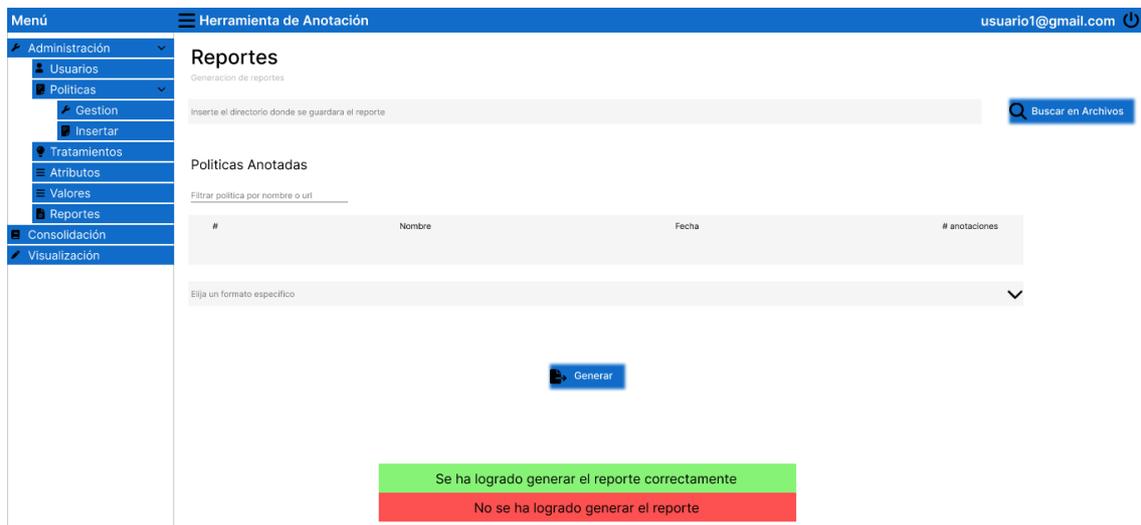


Figura 2.8. Diseño Interfaz Reportes.

2.2.4 DISEÑO SPRINT 4 – INTERNACIONALIZACIÓN

La internacionalización de la aplicación es el proceso que permite a la herramienta funcionar en múltiples escenarios. El alcance del trabajo es el cambio de idioma entre inglés y español. Se presenta el diseño general del *sprint* 4 que consta del diseño del diagrama de actividades y diseño de interfaces.

DISEÑO DE DIAGRAMA DE ACTIVIDADES

La Figura 2.9 presenta el diseño de diagramas de actividades para visualizar el funcionamiento del *sprint* la cual consiste en la internacionalización de la aplicación.

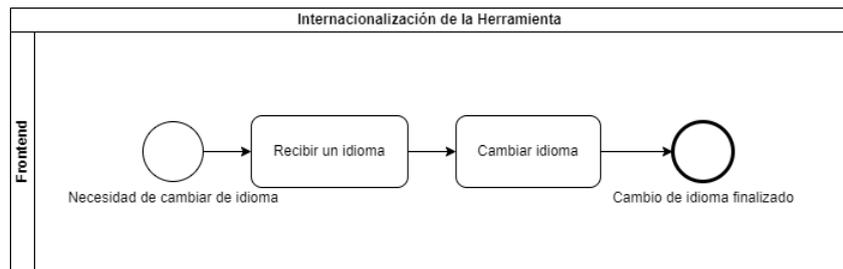


Figura 2.9. Diagrama de Actividades Internacionalización.

La internacionalización de la herramienta solo se aplicará a los contenidos del *frontend*, mientras que los textos recuperados del *backend* están fuera del alcance del proyecto. El *frontend* espera recibir una opción específica según la preferencia del usuario y cambia el idioma de los textos de la interfaz gráfica.

DISEÑO DE INTERFACES

En la Figura 2.10 se presenta el diseño de la interfaz para el cambio de idioma de la aplicación. Esta opción está disponible como un *select* en el *Header* de la herramienta.

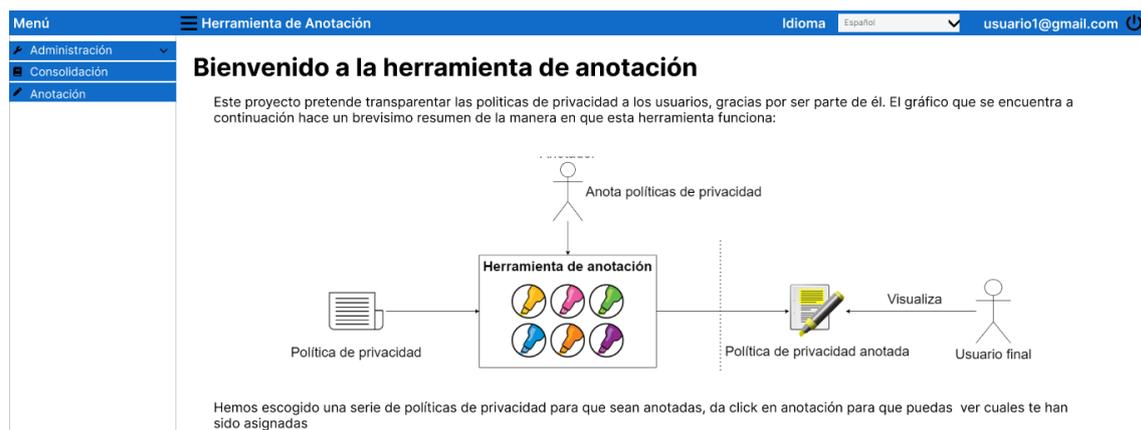


Figura 2.10. Diseño Interfaz Nueva Bienvenida.

2.3 IMPLEMENTACIÓN

En esta sección se presenta los aspectos más relevantes de la implementación de cada *sprint*.

2.3.1 IMPLEMENTACIÓN SPRINT 1 – DESPLIEGUE DEL ENTORNO DE DESARROLLO

Para la implementación de este *sprint* se han empleado las tecnologías detalladas en la Tabla 2.15. Estas tecnologías coinciden con aquellas usadas en el desarrollo de la primera versión de la herramienta.

Tabla 2.15. Tecnologías usadas durante el desarrollo de la herramienta

Componente	Lenguaje de programación	Versión	Framework	Versión	IDE
Backend	Python	3.8.6	Flask	1.1.2	Visual Studio Code
Frontend	TypeScript	3.5.3	Angular	8.2.7	Visual Studio Code
Servidor Linux Ubuntu	Bash	18	N/A	N/A	Terminal de Linux

2.3.2 IMPLEMENTACIÓN SPRINT 2 - EXTRACCIÓN DE POLÍTICAS

Este *Sprint* incluye la implementación de la funcionalidad para la extracción de políticas de privacidad individuales y grupales de forma automática.

IMPLEMENTACIÓN TAREAS: T-04 - SERVIDOR UBUNTU Y T-05 - DESPLIEGUE MICROSERVICIO

La implementación de estas tareas consiste en la incorporación y configuración del Servidor Linux con el microservicio. Como se mencionó en la sección 2.2.2 se ha elegido Linux Ubuntu 18, debido a que así lo recomienda el autor [15]. Se usó el software de virtualización Oracle VM VirtualBox para la instalación del sistema operativo.

Una vez instalado Linux Ubuntu 18 en VirtualBox, se procedió a descargar e instalar el microservicio. Como primer punto se resolvió los problemas de incompatibilidad de versiones del microservicio, esto es la actualización de chromedriver⁶ para Linux a la versión que se muestra en la Figura 2.11:

Index of /108.0.5359.71/

Name	Last modified	Size	Etag
Parent Directory			-
chromedriver_linux64.zip	2022-12-02 11:23:26	6.96MB	36fa81fb49b306330e8a496052008931
chromedriver_mac64.zip	2022-12-02 11:23:30	8.61MB	b6d423f668c68016e1e857da48c3ba47
chromedriver_mac_arm64.zip	2022-12-02 11:23:34	7.82MB	7e7a990dcb98c1fec78ba39d8da38bd9
chromedriver_win32.zip	2022-12-02 11:23:38	6.58MB	fd537a80faa26daecb09372c606a0c2c
notes.txt	2022-12-02 11:23:46	0.00MB	82ef3d2c25b0b0dae0bc0541cee2fa73

Figura 2.11. Chromedriver Actual

⁶ **Chromedriver:** servidor independiente que implementa el protocolo de cable de WebDriver para Chromium [21].

Para la comunicación del *backend* con el nuevo servidor se usó los servicios de *FTP* y *SSH*. *FTP* hace referencia al Protocolo de Transferencia de Archivos. Este tipo de protocolo es el estándar para transferir archivos entre dos ordenadores [16]. *SSH* (Secure Shell) es un protocolo de red que permite a los usuarios conectarse a un ordenador remoto y ejecutar programas en ese ordenador [16].

En el Código 2.1 se indican los comandos usados para la configuración de los servicios de FTP y SSH en el servidor Linux

```
1 apt install vsftpd
2 apt install openssh-server
```

Código 2.1. Configuración FTP y SSH.

IMPLEMENTACIÓN TAREA T-06 - AUTOMATIZACIÓN SERVIDOR LINUX

La tarea T-06 consiste en la creación de un Script de Linux para automatizar configuración del servidor y despliegue del microservicio. El Código 2.2 muestra el Script que configura el servidor y al mismo tiempo descarga el microservicio.

```
1 #!/bin/bash
2 apt update
3 apt -y upgrade
4 apt install rename
5 apt install vsftpd
6 apt install openssh-server
7 apt install -y openjdk-11-jdk
8 apt install -y wget
9 apt install unzip
10 apt install -y git
11 apt install -y python3
12 apt install -y python3-pip
13 pip3 install python-json-logger
14 pip3 install python3-pika
15 pip3 install tika
16 pip3 install selenium
17 pip3 install tld
18 pip3 install beautifulsoup4
19 pip3 install docx2txt
20 cd /tmp
21 wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
22 apt install -y ./google-chrome-stable_current_amd64.deb
23 wget https://chromedriver.storage.googleapis.com/108.0.5359.71/chromedriver_linux64.zip
24 unzip chromedriver_linux64.zip
25 apt install -y poppler-utils
26 mkdir /app
27 mv chromedriver /app/chromedriver
28 cd /app
29 #
30 git clone https://github.com/cesarmir1994/microservice3
31 cp -r microservice3/* /app
32 rm -r microservice3/
33 chmod 777 /app/*
```

Código 2.2. Script de Configuración Linux Ubuntu.

IMPLEMENTACIÓN TAREA T-07 - TREE-VIEW

La implementación de esta tarea consiste en la modificación del menú principal para mostrar los nuevos módulos en el *tree-view*. Debido a que los módulos son cargados directamente desde la base de datos, se creó dos nuevos componentes en Angular llamados *Insertar* y *Reportes* esto con el fin de modificar las migraciones con los nuevos módulos y los nuevos *paths* que se deben ejecutar al dar *click* en los mismos. El Código 2.3 muestra las nuevas líneas insertadas (12, 13 y 14), mientras que la Figura 2.12 muestra la base con los nuevos datos y también la interfaz.

```

1  op.bulk_insert(modulo_table,
2      [
3      {'id': 1,'nombre':'raiz','icono':'','path': None,'padre_id': None},
4      {'id': 2,'nombre':'Administracion','icono':'build','path': None,'padre_id':'1'},
5      {'id': 3,'nombre':'Consolidacion','icono':'book','path':'/paginas/consolidacion','padre_id':'1'},
6      {'id': 4,'nombre':'Anotacion','icono':'create','path':'/paginas/anotacion','padre_id':'1'},
7      {'id': 5,'nombre':'Usuarios','icono':'account_circle','path':'/paginas/administracion/usuarios','padre_id':'2'},
8      {'id': 6,'nombre':'Políticas','icono':'menu_book','path': None,'padre_id':'2'},
9      {'id': 7,'nombre':'Tratamientos','icono':'highlight','path':'/paginas/administracion/tratamientos','padre_id':'2'},
10     {'id': 9,'nombre':'Atributos','icono':'format_list_bulleted','path':'/paginas/administracion/atributos','padre_id':'2'},
11     {'id': 10,'nombre':'Valores','icono':'format_list_numbered','path':'/paginas/administracion/valores','padre_id':'2'},
12     {'id': 11,'nombre':'Reportes','icono':'insert_drive_file','path':'/paginas/administracion/reportes','padre_id':'2'},
13     {'id': 12,'nombre':'Gestion','icono':'create','path':'/paginas/administracion/politicas/gestion','padre_id':'6'},
14     {'id': 13,'nombre':'Insertar','icono':'note_add','path':'/paginas/administracion/politicas/insertar','padre_id':'6'},
15     ]
16 )

```

Código 2.3. Actualización de migraciones Tree-View.

id	nombre	icono	path	padre_id
1	raiz		[NULL]	[NULL]
2	Administracion	build	[NULL]	1
3	Consolidacion	book	/paginas/consolidacion	1
4	Anotacion	create	/paginas/anotacion	1
5	Usuarios	account_circle	/paginas/administracion/usuarios	2
6	Políticas	menu_book	[NULL]	2
7	Tratamientos	highlight	/paginas/administracion/tratamientos	2
9	Atributos	format_list_bulleted	/paginas/administracion/atributos	2
10	Valores	format_list_numbered	/paginas/administracion/valores	2
11	Reportes	insert_drive_file	/paginas/administracion/reportes	2
12	Gestion	create	/paginas/administracion/politicas/gestion	6
13	Insertar	note_add	/paginas/administracion/politicas/insertar	6



Figura 2.12. Actualización de módulos.

IMPLEMENTACIÓN TAREA T-08 - FRONTEND POLÍTICAS

La implementación de esta tarea consistió en la creación de la interfaz para la inserción de políticas de privacidad automáticamente. Como primer punto se mudó el componente que en la primera versión de la herramienta estaba alojado en el módulo *Políticas* hacia el módulo *Gestión*. En este módulo se puede ver, asignar e ingresar políticas manualmente, tal cual como estaba en la herramienta originalmente. En el módulo *Insertar* se ha programado la interfaz para recibir un archivo TXT. El Código 2.4 muestra el código HTML de la interfaz y la Figura 2.13 muestra el resultado [17].

```
1 <mat-card class="mat-elevation-z8 card" style="background-color: #fafafa ;">
2   <mat-grid-list cols="4" rowHeight="40px">
3     <mat-grid-tile [[colspan]="3" [[rowspan]="1">
4       <mat-card-title style="width: 100%"> {{ 'politica.insertar.titulo' | translate }}</mat-card-title>
5     </mat-grid-tile>
6     <mat-grid-tile [[colspan]="1" [[rowspan]="2">
7       <button mat-raised-button class="standar" style="height: 35px; color: white"
8         [(click)="informacion(true)"]>{{ 'politica.insertar.ayuda' | translate }}
9       <mat-icon>help</mat-icon>
10      </button>
11    </mat-grid-tile>
12    <mat-grid-tile [[colspan]="3" [[rowspan]="1">
13      <mat-card-subtitle style="width: 100%"> {{ 'politica.insertar.descripcion' | translate }}
14    </mat-card-subtitle>
15  </mat-grid-tile>
16  <mat-grid-tile [[colspan]="2"></mat-grid-tile>
17 </mat-grid-list>
18 <mat-card-content style="height: 75vh; overflow-y: auto; overflow-x: hidden;">
19   <mat-grid-list cols="10" rowHeight="40px">
20     <mat-grid-tile [[colspan]="9" [[rowspan]="2" style="padding-left: 15px;">
21       <mat-form-field class="example-full-width" appearance="outline">
22         <mat-label>{{ 'politica.insertar.archivo' | translate }}</mat-label>
23         <input id="nombreArchivo" [[disabled]="true" matInput placeholder="{{ 'politica.insertar.archivo' | translate
24 }}" value={{nombreArchivo}}>
25       </mat-form-field>
26     </mat-grid-tile>
27     <mat-grid-tile [[colspan]="1" [[rowspan]="1" style="padding-top: 17px;">
28       <button mat-raised-button class="standar" [(click)="subirArchivo()" Height="40px">
29         <mat-icon>file_copy</mat-icon>
30       </button>
31       <input id="archivo" type="file" accept=".txt" [(change)="escogerArchivo($event, target.files)"]>
32     </mat-grid-tile>
33   </mat-grid-list>
34   <div style="text-align: center" *ngIf="archivoEscogido">
35     <button mat-raised-button class="standar" [(click)="extraer()">
36       <mat-icon>archive</mat-icon>
37       {{ 'politica.insertar.extraer' | translate }}
38     </button>
39     <mat-card-subtitle style="width: 100%"></mat-card-subtitle>
40     <mat-card-subtitle style="width: 100%"> {{mensaje}}</mat-card-subtitle>
41     <mat-card-subtitle style="width: 100%"></mat-card-subtitle>
42   </div>
43   <div style="text-align: center" *ngIf="showSpinner">
44     <mat-card-subtitle style="width: 100%"></mat-card-subtitle>
```

```

45     <mat-card-subtitle style="width: 100%">/mat-card-subtitle>
46     <mat-grid-list cols="7" rowHeight="100px">
47         <mat-grid-tile colspan="7" rowspan="1">
48             <mat-spinner>/mat-spinner<
49         </mat-grid-tile>
50     </mat-grid-list>
51     <mat-card-subtitle style="width: 100%">/mat-card-subtitle>
52     <mat-card-subtitle style="width: 100%"> {{ 'politica.insertar.espere' | translate }}</mat-card-subtitle>
53 </div>
54 </mat-card-content>
</mat-card>

```

Código 2.4. HTML Políticas.

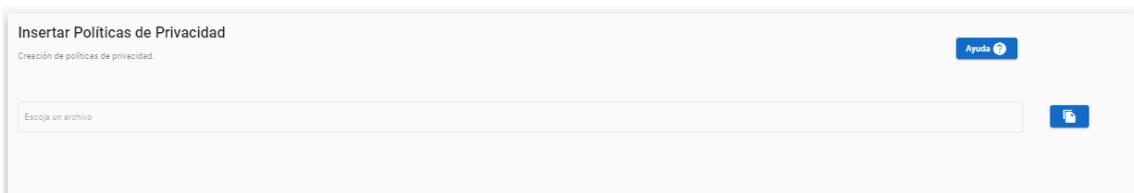


Figura 2.13. Interfaz Políticas.

IMPLEMENTACIÓN TAREA T-09 - LÓGICA FRONTEND POLÍTICAS

La tarea T-08 consistió en la creación de las funciones necesarios en el *frontend* para que el usuario pueda insertar una ruta con dirección a un archivo en formato TXT que contiene un conjunto de URLs y enviarlo al *backend*.

En la tarea anterior se programó el archivo HTML de tal forma que se pueda ingresar un archivo en formato TXT. En el archivo de Typescript de Angular se ha programado la función *consultarURLs* que envía el archivo al *backend* y este devuelve el número de líneas que posee el archivo. El Código 2.5 muestra la función *consultarURLs* y el Código 2.6 muestra como el *backend* cuenta las URLs y lo regresa al *frontend*.

```

1  consultarURLs(){
2      this._politicaService.contarUrls(this.archivoPolitica).subscribe(
3          resultado => {
4              let nota = '';
5              this.translate.stream("politica.insertar.num_url", {num: resultado.mensaje}).subscribe((res: string)=>{
6                  nota = res;
7              });
8              this.notificacion(nota, 'exito-snackbar')
9              this.mensaje = nota;
10             this.numero = +resultado.mensaje
11             this.archivoEscogido = true;
12         },
13         error => {
14             let nota = '';
15             this.translate.stream("politica.insertar.error_archivo").subscribe((res: string)=>{
16                 nota = res;
17             });
18             this.notificacion(nota, 'fracaso-snackbar');

```

```
19     this.mensaje = nota;
20     this.archivoEscogido = false;
21 }
22 )
23 }
```

Código 2.5. Consultar URLs Frontend.

```
1 #Cuenta las URLs de un archivo
2 def conteoURLs(nombre_archivo):
3     try:
4         documento = open(CARPETA_URL+"/"+nombre_archivo)
5         lineas = documento.readlines()
6         num = 0
7         for i in lineas:
8             if i != '\n':
9                 num = num+1
10        documento.close()
11        return num
12    except:
13        return "Error"
```

Código 2.6. Contar URLs Backend.

La Figura 2.14 muestra el archivo con el que se realizó la prueba, el cual contiene 2 URLs.



Figura 2.14. Archivo de Texto.

La Figura 2.15 muestra el ingreso del archivo *Políticas.txt*. Además, se indica que la aplicación solo admite archivos en formato TXT.

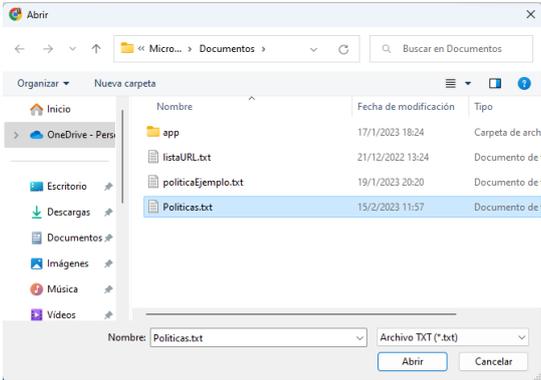


Figura 2.15. Ingreso de Archivo de Texto.

Si todo ha salido bien con el archivo TXT se desbloquea un botón llamado *Extraer*. La Figura 2.16 muestra el resultado obtenido.



Figura 2.16. Resultado de Leer Archivo de Texto.

IMPLEMENTACIÓN TAREA T-10 - CONEXIÓN BACKEND Y SERVIDOR LINUX

La tarea T-10 consistió en la creación de un script en Python para la conexión entre el *backend* de la herramienta de anotación con el nuevo servidor Ubuntu y en el envío del archivo en formato TXT. En esta parte se hace uso de los servicios FTP y SSH instalados previamente en el servidor. Primero se crea una conexión SSH con la dirección IP y una clave. Luego, se entra en el servidor y se edita el archivo *listaURL.txt* con la nueva URL. Se ejecuta el microservicio y se abre una conexión FTP para descargar el archivo generado por el microservicio. Finalmente se envía el texto de la política y la URL que se usó al *frontend*. El Código 2.7 muestra la programación del Script.

```

1 #Extrae el texto de políticas de privacidad desde el servidor ubuntu
2 def extraerTexto(data):
3     #try:
4     cliente = paramiko.SSHClient()
5     cliente.set_missing_host_key_policy(paramiko.AutoAddPolicy())
6     cliente.connect(servidor,username=usuario,password=clave)
7     archivo = cliente.open_sftp()
8     #Creamos el archivo con 1 URL
9     url = encuentraURL(data['num'])
10    if url!='Error':
11        if os.path.exists(CARPETA_URL+"/listaURL.txt"):
12            os.remove(CARPETA_URL+"/listaURL.txt")
13        if os.path.exists(CARPETA_SUBIDA+"/PrivacyPolicy1.txt"):
14            os.remove(CARPETA_SUBIDA+"/PrivacyPolicy1.txt")
15        file = open(CARPETA_URL+"/listaURL.txt", "w")
16        file.write(url)
17        file.close()
18    else:
19        respuesta = {
20            'estado': 'fallido',
21            'mensaje': 'No se logro crear el archivo con la URL'
22        }
23        return respuesta, 409
24    #Se envia el archivo txt a linux
25    archivo.put(CARPETA_URL+"/listaURL.txt", "/app/listaURL.txt")
26    #Se ejecuta el microservicio3 en linux
27    stdin, stdout, stderr = cliente.exec_command(
28        "cd /app ; echo s | rm -r result/* ; echo 4652 | sudo -S python3 micro3-g.py"
29    )

```

```

30 #Se espera el tiempo suficiente segun el numero de URLs
31 print("Descargando archivos, por favor espere...")
32 time.sleep(30)
33 resultado = stdout.read().decode()
34 print(resultado)
35 #Se guarda en una cadena los archivos descargados y se los cambia de nombre
36 entrada, salida, error = cliente.exec_command(
37     "cd /app; sh CambiarNombres.sh;cd /app/result; ls"
38 )
39 time.sleep(1)
40 res = salida.read().decode()
41 #Se filtran las cadenas que terminan en .txt
42 secuencia = r'\w*.txt'
43 busqueda = re.findall(secuencia,res)
44 #Se descargan los archivos .txt
45 ruta = "/app/result/"
46 for i in busqueda:
47     path_origen = ruta + i
48     path_destino = CARPETA_SUBIDA+i
49     archivo.get(path_origen,path_destino)
50 #Se eliminan del servidor los archivos
51 entrada2, salida2, error2 = cliente.exec_command(
52     "cd /app;rm -rf result;mkdir result"
53 )
54 time.sleep(1)
55 archivo.close()
56 cliente.close()
57 politica = abrir_politica("PrivacyPolicy1.txt")
58 if os.path.exists(CARPETA_SUBIDA+"/PrivacyPolicy1.txt"):
59     respuesta = {
60         'estado': 'exito',
61         'url': url,
62         'texto_politica': politica,
63     }
64     return respuesta, 201
65 else:
66     respuesta = {
67         'estado': 'fallido',
68         'mensaje': 'No se logro crear el archivo con la URL'
69     }
70     return respuesta, 409

```

Código 2.7. Conexión Backend y Servidor Linux.

IMPLEMENTACION TAREA T-11 - MODIFICACIÓN MICROSERVICIO

La implementación de la tarea T-11 consiste en la adaptación y modificación del microservicio hacia las nuevas necesidades. Debido a la naturaleza del Script creado en la tarea anterior, se debe modificar ciertos parámetros del microservicio. La Tabla 2.16 muestra las modificaciones que se llevaron a cabo.

Tabla 2.16. Modificaciones del microservicio.

Microservicio 3		
#	Original	Modificación
1	Espera a que el usuario ingrese una ruta de un archivo TXT donde están las URLs	Hay un archivo TXT quemado al mismo nivel que el código del microservicio, así que siempre tomará ese archivo como input.
2	El nombre del archivo descargado depende de varios parámetros	El nombre del archivo siempre será <i>PrivacyPolicy1</i> y el nombre final lo dará el usuario.
3	Descarga archivos en formato .docx, PDF, HTML y TXT	Descarga solo archivos en formato TXT

El Código 2.8 muestra la modificación del ítem 1 de la Tabla 2.16, en el que se modificó el código fuente del microservicio llamado *micro3-g.py*.

```

1 def Service3():
2     print('Entering microservice 3')
3     path = 'listaURL.txt'
4     elements = apk_list(path)

```

Código 2.8. Modificación Ítem 1.

El Código 2.9 muestra la modificación del ítem 2 de la Tabla 2.15. Se creó un Script llamado *CambiarNombre.sh* para que realice este proceso.

```

1 #!/bin/bash
2 aux=1
3 #Quita espacios en blanco
4 rename 's/\s/_/g' /app/result/*.txt
5 #Renombra los txt
6 for i in $( find /app/result -type f -iname "*.txt" )
7 do
8     mv $i /app/result/PrivacyPolicy$aux.txt
9     aux=$((aux+1))
10 done

```

Código 2.9. Modificación Ítem 2.

La modificación del ítem 3 lo realiza el Código 2.7, que se implementó en la tarea 10.

IMPLEMENTACIÓN TAREA T-12 – VALIDACION FORMULARIO POLÍTICAS

La tarea T-12 consiste en la generación de un formulario en el *frontend* para insertar las URLs de la o las políticas de privacidad a descargar y también para modificar los textos descargados. También aquí se creará lógica necesaria en el *backend* para guardar las nuevas políticas de privacidad en la base de datos.

El Código 2.10 muestra la programación en HTML del formulario. La tarea T-10 es la encargada de enviar el texto de la política de privacidad y la URL. Estos datos se los ordenara en los inputs del formulario según correspondan. El usuario podrá ingresar: nombre de la política, fecha y editar el texto, la URL no se podrá modificar. La Figura 2.17 muestra el resultado.

```

1 <div>
2   <mat-toolbar class="standar">
3     <span style="width: 100%;text-align: center;">{{ 'politica.insertar.dialogo.titulo' | translate }}</span>
4     <button mat-raised-button class="standar" style="height: 35px; color: white"
5       [(click)="mostrarAyuda(false)"]>
6       <mat-icon>help</mat-icon>
7     </button>
8   </mat-toolbar>
9   <mat-card>
10    <mat-card-content>
11      <mat-dialog-content [formGroup]="formulario">
12        <mat-form-field class="example-full-width" appearance="outline">
13          <mat-label>{{ 'politica.gestion.dialogo.nombre' | translate }}</mat-label>
14          <input id="nombrePolitica" type="text" matInput
15            placeholder="{{ 'politica.gestion.dialogo.nombre' | translate }}"
16            formControlName="nombre" value="{{politicaAux.nombre}} required">
17          <mat-error [ngIf]="nombre.invalid">{{errorNombre()}}</mat-error>
18        </mat-form-field>
19        <mat-form-field class="example-full-width" appearance="outline">
20          <mat-label>{{ 'politica.gestion.dialogo.url' | translate }}</mat-label>
21          <input id="urlPolitica" matInput placeholder="{{ 'politica.gestion.dialogo.url' | translate }}"
22            formControlName="url" value="{{politicaAux.url}} required readonly">
23          <mat-error [ngIf]="url.invalid">{{errorUrl()}}</mat-error>
24        </mat-form-field>
25        <mat-form-field class="example-full-width" appearance="outline">
26          <mat-label>{{ 'politica.gestion.dialogo.fecha' | translate }}</mat-label>
27          <input id="fechaPolitica" matInput
28            [matDatepicker]="picker">
29            placeholder="{{ 'politica.gestion.dialogo.fecha' | translate }}"
30            formControlName="fecha"
31            required>
32          <mat-datepicker-toggle matSuffix [for]="picker"></mat-datepicker-toggle>
33          <mat-datepicker [picker]></mat-datepicker>
34          <mat-error [ngIf]="url.invalid">{{errorFecha()}}</mat-error>
35        </mat-form-field>
36        <mat-form-field class="example-full-width" appearance="outline">
37          <mat-label>{{ 'politica.insertar.dialogo.texto' | translate }}</mat-label>
38          <textarea id="politicaTexto" style="height: 200px;" matInput
39            placeholder="{{ 'politica.insertar.dialogo.texto' | translate }}"></textarea>
40          <mat-error [ngIf]="aux">{{errorContenido()}}</mat-error>
41        </mat-form-field>
42        <mat-grid-list cols="12" rowHeight="40px">
43          <mat-grid-tile [colspan]="2" [rowspan]="1">
44            <button mat-raised-button color="warn" Height="40px" [(click)="mostrarConfirmacion(0)">
45              <mat-icon>cancel</mat-icon>
46              {{ 'botones.cancelar' | translate }}
47            </button>
48          </mat-grid-tile>
49          <mat-grid-tile [colspan]="2" [rowspan]="1" >

```

```

50         <button mat-raised-button color="accent" Height="40px" [(click)="mostrarConfirmacion(1)">
51             <mat-icon>save</mat-icon>
52             {{ 'botones.guardar' | translate }}
53         </button>
54     </mat-grid-tile>
55     <mat-grid-tile [colspan]="5" [rowspan]="1" ></mat-grid-tile>
56     <mat-grid-tile [colspan]="3" [rowspan]="1" >
57         <button mat-raised-button class="standar" Height="40px" [(click)="mostrarConfirmacion(2)">
58             <mat-icon>arrow_forward</mat-icon>
59             {{ 'botones.siguiente' | translate }}
60         </button>
61         <input type="file" accept=".txt" >
62     </mat-grid-tile>
63 </mat-grid-list>
64 </mat-dialog-content>
65 </mat-card-content>
66 </mat-card>
67 </div>

```

Código 2.10. HTML Formulario Políticas.

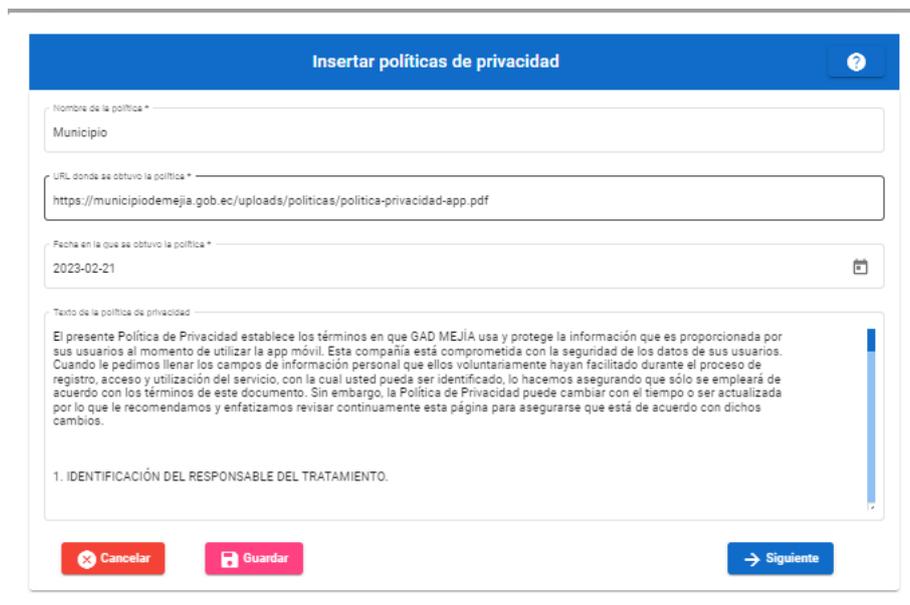


Figura 2.17. Resultado de Extracción de Texto.

El usuario está en la capacidad de decidir si guarda la política de privacidad, cancela la operación o avanza a la siguiente URL. También se ha creado una ayuda para dar formato al texto de la política de privacidad, el usuario puede acceder a la ayuda pulsando el botón de la parte superior derecha. La Figura 2.18 muestra la interfaz de ayuda.

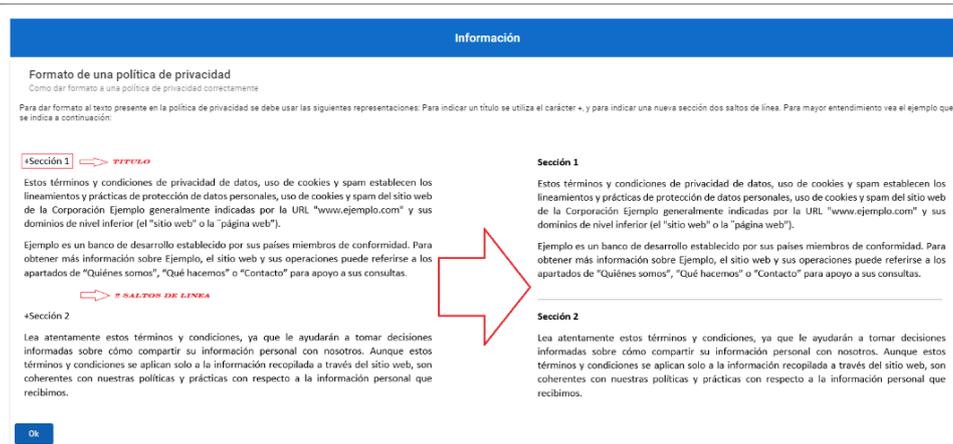


Figura 2.18. Ayuda Formato de Texto de Política de Privacidad.

La lógica para guardar la política de privacidad en la base de datos es muy similar a la usada en la primera versión de la herramienta, por tal no es necesario volver a explicarlo en este documento.

2.3.3 IMPLEMENTACIÓN SPRINT 3 – REPORTE

Este *sprint* incluye la implementación de un módulo para la generación de reportes de políticas de privacidad anotadas de forma automática.

IMPLEMENTACIÓN TAREA T-13 - CONSULTA SQL

La implementación de esta tarea consiste en la creación de una consulta SQL usando SQLAlchemy, con el fin de obtener los datos necesarios para realizar reportes de políticas anotadas.

El Código 2.11 muestra la creación de una clase llamada *PoliticaAnotadaReporte*, se la crea con el objetivo de poder llamar una variable de este tipo y guardar los datos de la consulta.

```

1 class PoliticaAnotadaReporte:
2     def __init__(self, politica_id, politica_nombre, url, parrafo_id, parrafo_texto, anotacion_id, fecha_annotado,
3                 permite, anotacion_texto, valor_id, valor, atributo_id, atributo, tratamiento_id, tratamiento):
4         self.politica_id = politica_id
5         self.politica_nombre = politica_nombre
6         self.url = url
7         self.parrafo_id = parrafo_id
8         self.parrafo_texto = parrafo_texto
9         self.anotacion_id = anotacion_id
10        self.fecha_annotado = fecha_annotado
11        self.permite = permite
12        self.anotacion_texto = anotacion_texto
13        self.valor_id = valor_id
14        self.valor = valor
15        self.atributo_id = atributo_id
16        self.atributo = atributo
17        self.tratamiento_id = tratamiento_id

```

Código 2.11. Clase PoliticaAnotadaReporte.

EL Código 2.12 muestra la programación de la consulta usando SQLAlchemy. Los datos de la consulta son guardados en un array de tipo *PoliticaAnotadaReporte*. Se debe enviar como parámetros el *idPolitica*, *idUsuario* y *consolidar*, este último hace referencia a que si el usuario ha realizado la anotación como anotador o consolidador.

```

1 #Genera reportes de politicas anotadas
2 def generarReporte(data):
3     politicas_annotadas = []
4     for item in data['politicas']:
5         if(item['tipo_usuario']=='Consolidador'):
6             item['tipo_usuario']=1
7         else:
8             item['tipo_usuario']=0
9     consulta = (db.session.query(
10         Anotacion.id, Anotacion.fecha_annotado, Anotacion.ejecuta, Anotacion.texto,
11         Politica.id, Politica.nombre, Politica.url,
12         Parrafo.id, Parrafo.texto,
13         Valor.id, Valor.descripcion,
14         Atributo.id, Atributo.descripcion,
15         Tratamiento.id, Atributo.descripcion)
16         .outerjoin(Parrafo, Parrafo.id == Anotacion.parrafo_id)
17         .outerjoin(Politica, Politica.id == Parrafo.politica_id)
18         .outerjoin(AnotacionValorRelacion, AnotacionValorRelacion.anotacion_id == Anotacion.id)
19         .outerjoin(Valor, Valor.id == AnotacionValorRelacion.valor_id)
20         .outerjoin(Atributo, Atributo.id == Valor.atributo_id)
21         .outerjoin(Tratamiento, Tratamiento.id == Atributo.tratamiento_id)
22         .filter(Politica.id == item['id_politica'],
23             Anotacion.usuario_id == item['id_usuario'],
24             Anotacion.consolidar == item['tipo_usuario']).all() )
25     if not consulta:
26         print('No hay datos')
27     else:
28         for items in consulta:
29             aux = ''
30             aux = PoliticaAnotadaReporte(items[4],items[5],items[6],items[7],items[8],
31                 items[0],items[1],items[2],items[3],items[9],
32                 items[10],items[11],items[12],items[13],items[14])
33             politicas_annotadas.append(aux)

```

Código 2.12. Consulta SQL Reportes.

IMPLEMENTACIÓN TAREA T-14 - REPORTES EN CSV

La tarea T-14 consiste en la creación de un Script de Python para generar reportes de políticas anotadas en formato CSV. Una vez obtenido el array con todos los datos necesarios, se usa la librería CSV de Python para abrir un archivo CSV y escribir la información dentro. El nombre del reporte será *Reporte+{fecha}*

El Código 2.13 muestra la creación del archivo CSV.

```

1 fecha_actual = datetime.datetime.now()
2 aux2 = fecha_actual.strftime('%Y-%m-%d-%H-%M-%S')
3 if (data['formato'] == 'csv'):
4     nombre_archivo = "Reporte-"+aux2+".csv"
5     if os.path.exists(CARPETA_REPORTE+"/"+nombre_archivo):
6         os.remove(CARPETA_REPORTE+"/"+nombre_archivo)
7     with open(CARPETA_REPORTE+"/"+nombre_archivo,'w',newline='',encoding='utf-8') as csvfile:
8         writer = csv.DictWriter(csvfile,fieldnames=[
9             'politica_id', 'politica_nombre', 'url','parrafo_id','parrafo_texto','anotacion_id',
10            'fecha_annotado','permite','anotacion_texto','valor_id','valor','atributo_id','atributo',
11            'tratamiento_id','tratamiento'])
12        writer.writeheader()
13        for items in politicas_annotadas:
14            writer.writerow({
15                'politica_id':items.politica_id, 'politica_nombre':items.politica_nombre,
16                'url':items.url, 'parrafo_id':items.parrafo_id, 'parrafo_texto':items.parrafo_texto,
17                'anotacion_id':items.anotacion_id, 'fecha_annotado':items.fecha_annotado, 'permite':items.permite,
18                'anotacion_texto':items.anotacion_texto, 'valor_id':items.valor_id, 'valor':items.valor,
19                'atributo_id':items.atributo_id, 'atributo':items.atributo, 'tratamiento_id':items.tratamiento_id,
20                'tratamiento':items.tratamiento})
21        reportes = open(CARPETA_REPORTE+"/"+nombre_archivo, 'r')
22        respuesta = {
23            'estado': 'exito',
24            'mensaje': 'Se ha generado el reporte exitosamente',
25            'reporte': marshal(reportes, PoliticaDto.reporte)
26        }
27        return respuesta, 201

```

Código 2.13. Creación Archivo SCV.

IMPLEMENTACIÓN TAREA T-15 - REPORTES EN YAML

La tarea T-15 consiste en la creación de un Script de Python para generar reportes de políticas anotadas en formato YAML. Una vez obtenido el array con todos los datos necesarios, se usa la librería YAML de Python para abrir un archivo YAML y escribir la información dentro. Se crea un reporte por cada política que el *frontend* envíe. Los datos se escriben por partes y en diferentes funciones. El nombre del reporte será *Reporte+{nombre de política}+{fecha}*

El Código 2.14 muestra la creación del archivo YAML.

```

1 if (data['formato'] == 'yaml'):
2     obtenerPolíticas(políticas_annotadas, aux2)
3     respuesta = {
4         'estado': 'exito',
5         'mensaje': 'Se ha generado el reporte exitosamente',
6     }
7     return respuesta, 201
8
9 #Se obtiene la informacion para reporte en formato YAML
10 #Políticas para YAML
11 def obtenerPolíticas(políticasAnotadas, fecha):
12     data = {}
13     idPolíticas = []

```

```

14     i=1
15     for aux in politicasAnotadas:
16         if aux.politica_id not in idPoliticass:
17             idPoliticass.append(aux.politica_id)
18             data1 = { "policy_"+str(i) : {
19                 "policy_id": aux.politica_id,
20                 "policy_name" : aux.politica_nombre,
21                 "policy_type" : "TEST",
22                 "contains_synthetic": False,
23                 "segments" : obtenerParrafos(aux.politica_id, politicasAnotadas),
24             }}
25             generar_yaml(data1, aux.politica_nombre, fecha)
26             #data.update(data1)
27             i=i+1
28     return data
29
30 #Parrafo para YAML
31 def obtenerParrafos(idPolitica, politicasAnotadas):
32     data = {}
33     i=1
34     for a in politicasAnotadas:
35         if(idPolitica == a.politica_id):
36             data1 = {
37                 "segments_"+str(i):{
38                     "segment_id" : a.parrafo_id,
39                     "segment_text" : a.parrafo_texto,
40                     "performed" : a.permite,
41                     "sentences" : obtenerAnotaciones(a.politica_id, a.parrafo_id, politicasAnotadas)
42                 }}
43             data.update(data1)
44             i=i+1
45     return(data)
46
47 #Anotaciones para YAML
48 def obtenerAnotaciones(idPolitica, idParrafo, politicasAnotadas):
49     data = {}
50     i=1
51     for a in politicasAnotadas:
52         if(idPolitica == a.politica_id and idParrafo == a.parrafo_id):
53             data1 = {
54                 "sentences_"+str(i):{
55                     "sentence_text": a.anotacion_texto,
56                     "practice_type": a.tratamiento,
57                     "practice_att": a.atributo,
58                     "practice_value": a.valor,
59                 }}
60             data.update(data1)
61             i=i+1
62     return(data)
63
64 def generar_yaml(data, nombre, fecha):
65     nombre_archivo = "Reporte-"+nombre+"-"+fecha+".yaml"
66     if os.path.exists(CARPETA_REPORTES+"/"+nombre_archivo):
67         os.remove(CARPETA_REPORTES+"/"+nombre_archivo)
68     with open(CARPETA_REPORTES+"/"+nombre_archivo, 'w', encoding='utf-8') as f:
69         yaml.dump(data, f, default_flow_style=False)

```

Código 2.14. Creación de archivo YAML.

IMPLEMENTACIÓN TAREA T-16 - TABLA DE ANOTACIONES

La tarea T-16 consiste en la generación de una interfaz gráfica que muestre una tabla con las políticas anotadas. Para mostrar la tabla primero el usuario debe escoger el tipo de políticas anotadas. Estas se dividen en dos grupos: Finalizadas y No Finalizadas. En la tabla se despliega el nombre de la política, el número de anotaciones, el nombre del usuario que anoto, el tipo de usuario (anotador o consolidador) y un *CheckBox* para escoger la política.

El Código 2.15 muestra el archivo HTML donde se crea un *Select* para escoger el tipo de políticas y la tabla para albergar las políticas.

```
1 <mat-card class="mat-elevation-z8 card" style="background-color: #fafafa ;">
2   <mat-grid-list cols="4" rowHeight="40px">
3     <mat-grid-tile [[colspan]="3" [[rowspan]="1">
4       <mat-card-title style="width: 100%">{{ 'reporte.titulo' | translate }}</mat-card-title>
5     </mat-grid-tile>
6     <mat-grid-tile [[colspan]="3" [[rowspan]="1">
7       <mat-card-subtitle style="width: 100%"> {{ 'reporte.descripcion' | translate }}
8     </mat-card-subtitle>
9     </mat-grid-tile>
10    <mat-grid-tile [[colspan]="2"></mat-grid-tile>
11  </mat-grid-list>
12  <mat-dialog-content [[formGroup]="reporteFormulario">
13    <mat-grid-list cols="6" rowHeight="40px" [[ngStyle]="{'background-color': '#fafafa'}">
14      <mat-grid-tile [[colspan]="1" [[rowspan]="2">
15        </mat-grid-tile>
16      <mat-grid-tile [[colspan]="4" [[rowspan]="2">
17        <mat-form-field class="example-full-width" appearance="outline">
18          <mat-label>{{ 'reporte.politica' | translate }}</mat-label>
19          <mat-select [(value)]="tipo2" FormControlName="tipo">
20            <mat-option style="color: black; font-weight: bold;">
21              {{ 'reporte.ninguno' | translate }}
22            </mat-option>
23            <mat-option value="finalizadas" (click)="desplegarPoliticAnotadas(tipo2)">
24              {{ 'reporte.tipo.finalizadas' | translate }}
25            </mat-option>
26            <mat-option value="noFinalizadas" (click)="desplegarPoliticAnotadas(tipo2)">
27              {{ 'reporte.tipo.no_finalizadas' | translate }}
28            </mat-option>
29          </mat-select>
30          <mat-error [[ngIf]="formato.invalid">{{errorTipo()}}</mat-error>
31        </mat-form-field>
32      </mat-grid-tile>
33      <mat-grid-tile [[colspan]="1" [[rowspan]="2">
34        </mat-grid-tile>
35      </mat-grid-list>
36      <mat-card-content style="height: 75vh; overflow-y: auto; overflow-x: hidden;">
37        <mat-form-field style="min-width: 210px;">
38          <input matInput [(keyup)]="aplicarFiltro($event,target,value)"
39            placeholder="{{ 'reporte.filtro' | translate }}">
40        </mat-form-field>
```

```

41     <table mat-table [dataSource]="dataSource" class="mat-elevation-z8" style="width: 100%">
42       <tr mat-header-row *matHeaderRowDef="displayedColumns; sticky: true"></tr>
43       <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
44       <ng-container matColumnDef="id">
45         <th mat-header-cell *matHeaderCellDef> # </th>
46         <td mat-cell *matCellDef="let politica; let indice = index"> {{indice+1}} </td>
47       </ng-container>
48       <ng-container matColumnDef="nombre" >
49         <th mat-header-cell *matHeaderCellDef> {{ 'reporte.tabla.nombre' | translate }}</th>
50         <td mat-cell *matCellDef="let politica">
51           {{politica.nombre}}
52         </td>
53       </ng-container>
54       <ng-container matColumnDef="usuario">
55         <th mat-header-cell *matHeaderCellDef> {{ 'reporte.tabla.usuario' | translate }} </th>
56         <td mat-cell *matCellDef="let politica">
57           {{politica.usuario}}
58         </td>
59       </ng-container>
60       <ng-container matColumnDef="tipo_usuario">
61         <th mat-header-cell *matHeaderCellDef> {{ 'reporte.tabla.tipo' | translate }} </th>
62         <td mat-cell *matCellDef="let politica">
63           {{politica.tipo_usuario}}
64         </td>
65       </ng-container>
66       <ng-container matColumnDef="fecha">
67         <th mat-header-cell *matHeaderCellDef> {{ 'reporte.tabla.fecha' | translate }} </th>
68         <td mat-cell *matCellDef="let politica">
69           {{politica.fecha}}
70         </td>
71       </ng-container>
72       <ng-container matColumnDef="anotaciones">
73         <th mat-header-cell *matHeaderCellDef> {{ 'reporte.tabla.numero' | translate }} </th>
74         <td mat-cell *matCellDef="let politica">
75           {{politica.anotaciones}}
76         </td>
77       </ng-container>
78       <ng-container matColumnDef="escoger">
79         <th mat-header-cell *matHeaderCellDef> {{ 'reporte.tabla.escoger' | translate }}
80         <input *ngIf="aux" type="checkbox" [(change)]=seleccionarTodo($event,target,checked)"
81           class="form-checkbox h-5 w-5 text-indigo-600" [checked]="encabezado">
82         </th>
83         <td mat-cell *matCellDef="let politica">
84           <input type="checkbox" [(change)]=onChange(politica,$event,target,checked)"
85           class="form-checkbox h-5 w-5 text-indigo-600" [checked]="politica.escoger">
86         </td>
87       </ng-container>
88     </table>

```

Código 2.15. HTML Reportes.

El Código 2.16 muestra el archivo Typescript donde se llama al *backend* para consultar las políticas.

```

1  consultarPoliticAnotadas(tipo: number){

```

```

2  this.checkbox = false;
3  this.checkboxError=false;
4  this.politicasReporte = [];
5  this._politicaService.consultarPoliticasAnotadas(tipo).subscribe(
6  resultado => {
7      console.log(resultado)
8      this.dataSource = new MatTableDataSource(resultado),
9      this.dataSource.filterPredicate = function(data, filter : string): boolean {
10         return data.nombre.toLowerCase().includes(filter)
11     }
12     this.numRegistros=this.dataSource.data.length;
13     this.aux = true;
14 },
15 () => {
16     let nota = '';
17     this.translate.stream("reporte.error_politica").subscribe((res: string)=>{
18         nota = res;
19     });
20     this.notificacion(nota, "fracaso-snackbar", 4000)
21 }
22
23

```

Código 2.16. Consultar Políticas Anotadas Frontend.

El Código 2.17 muestra al *backend* creando la consulta, la cual se divide en 3: consulta de datos básicos, consulta de anotaciones y consulta de última fecha de anotación.

```

1  #Consulta todas las politicas de privacidad anotadas
2  def consultar_politicas_annotadas(tipo):
3      #Tipo de politicas a consultar
4      #0 --> No finalizadas
5      #1 --> Finalizadas
6      if(tipo=='0'):
7          finalizado = 0
8      else:
9          finalizado = 1
10     return politicas_annotadas(finalizado)
11
12     def politicas_annotadas(finalizado):
13         politicas_annotadas = []
14         consulta = (db.session.query(Politica.id,Usuario.id,Politica.nombre, Usuario.email, PoliticaUsuarioRelacion.consolidar)
15             .join(Politica, Politica.id == PoliticaUsuarioRelacion.politica_id)
16             .join(Usuario, Usuario.id == PoliticaUsuarioRelacion.usuario_id)
17             .filter(PoliticaUsuarioRelacion.finalizado == finalizado).all() )
18         if not consulta:
19             respuesta = {
20                 'estado': 'fallido',
21                 'mensaje': 'No existen politicas anotadas'
22             }
23             return respuesta, 409
24         else:
25             for items in consulta:
26                 if(items[4]==True):
27                     tipo_usuario='Consolidador'

```

```

28     else:
29         tipo_usuario='Anotador'
30         fecha = consultar_fecha(items[0],items[1], items[4])
31         anotaciones = consultar_anotaciones(items[0],items[1], items[4])
32         aux = PoliticaAnotadaConsultar(items[0], items[1],items[2], items[3], tipo_usuario, fecha, anotaciones)
33         politicas_annotadas.append(aux)
34     return marshal(politicas_annotadas, PoliticaDto.politicaAnotadaConsultar), 201
35
36 #Consulta la ultima fecha de anotacion de una politica
37 def consultar_fecha(politica_id, usuario_id, tipo_usuario):
38     fecha = (db.session.query(Anotacion.fecha_annotado)
39             .outerjoin(Farrafo, Anotacion.parrafo_id == Farrafo.id)
40             .outerjoin(Politica, Politica.id == Farrafo.politica_id)
41             .filter(Politica.id == politica_id, Anotacion.usuario_id == usuario_id, Anotacion.consolidar==tipo_usuario)
42             .order_by(Anotacion.fecha_annotado)).first()
43     if not fecha:
44         fecha = None
45     else:
46         fecha = fecha[0]
47     return fecha
48
49 #Consulta numero de anotacion de una politica
50 def consultar_anotaciones(politica_id, usuario_id, tipo_usuario):
51     anotaciones = (db.session.query(Anotacion)
52                   .outerjoin(Farrafo, Anotacion.parrafo_id == Farrafo.id)
53                   .outerjoin(Usuario, Usuario.id == Anotacion.usuario_id)
54                   .filter(Farrafo.politica_id == politica_id, Anotacion.usuario_id == usuario_id, Anotacion.consolidar == tipo_usuario)
55                   .count())
56     if not anotaciones:
57         anotaciones = 0
58     return anotaciones

```

Código 2.17. Consultar Políticas Anotadas Backend.

La Figura 2.19 muestra el resultado de la tarea finalizada.

Reportes
Generación de reportes de políticas de privacidad anotadas.

escoge el tipo de políticas anotadas
Finalizadas

Filtrar política por nombre

#	Nombre Política	Usuario	Tipo de Usuario	Fecha de última Anotación	Número de Anotaciones	Escoger <input type="checkbox"/>
1	Firefly	admin@gmail.com	Anotador	2023-01-18	7	<input type="checkbox"/>
2	SM	admin@gmail.com	Anotador	2023-01-19	4	<input type="checkbox"/>
3	Nike	admin@gmail.com	Consolidador	2023-01-21	6	<input type="checkbox"/>
4	Nike	admin@gmail.com	Anotador	2023-01-21	6	<input type="checkbox"/>
5	Firefly	david@gmail.com	Anotador	2023-01-19	7	<input type="checkbox"/>
6	Nike	david@gmail.com	Anotador	2023-01-21	6	<input type="checkbox"/>

Figura 2.19. Interfaz Reportes.

IMPLEMENTACIÓN TAREA T-17 - FORMULARIO REPORTES

La tarea T-17 consiste en la creación y validación de formulario para escoger las políticas anotadas y el formato del reporte. Los formatos disponibles son CSV y YAML.

El Código 2.18 muestra el complemento del HTML de la tarea anterior.

```
1 <mat-grid-list cols="6" rowHeight="40px" [ngStyle]="{'background-color': '#fafafa'}">
2   <mat-grid-tile [colspan]="1" [rowspan]="2">
3     </mat-grid-tile>
4   <mat-grid-tile [colspan]="4" [rowspan]="2">
5     <mat-form-field class="example-full-width" appearance="outline">
6       <mat-label>{{ 'reporte.formato' | translate }} </mat-label>
7       <mat-select [(value)]="formato2" formControlName="formato" required>
8         <mat-option style="color: black; font-weight: bold;">
9           {{ 'reporte.ninguno' | translate }}
10        </mat-option>
11        <mat-option value="yaml" [(click)="formatoReporte(formato2)">YAML </mat-option>
12        <mat-option value="csv" [(click)="formatoReporte(formato2)">CSV </mat-option>
13      </mat-select>
14      <mat-error [ngIf]="formato.invalid">{{errorFormato()}} </mat-error>
15    </mat-form-field>
16  </mat-grid-tile>
17  <mat-grid-tile [colspan]="1" [rowspan]="2">
18    </mat-grid-tile>
19 </mat-grid-list>
20 <div style="text-align: center">
21   <button mat-raised-button class="standar" [disabled]="reporteFormulario.invalid || !checkbox">
22     [(click)="generarReporte()]">
23     <mat-icon>move_to_inbox</mat-icon>
24     {{ 'botones.reporte' | translate }}
25   </button>
26 </div>
27 <div style="text-align: center">
28   <mat-card-subtitle style="width: 100%; height: 35px;"></mat-card-subtitle>
29 </div>
```

Código 2.18. HTML Formulario Reportes.

El Código 2.19 muestra las validaciones del formulario en el archivo Typescript. Una vez se cumpla las validaciones se desbloquea el botón que permite generar el reporte.

```
errorFormato() {
  let nota = '';
2  this.translate.stream("reporte.formato").subscribe((res: string)=>{
3    nota = res;
4  });
5  return this.formato.hasError("required") ? nota : "";
6 }
7
8 errorCheckbox(){
9  let nota = '';
10  this.translate.stream("reporte.error_escoger").subscribe((res: string)=>{
11    nota = res;
12  });
13  return this.formato.hasError("required") ? nota : "";
14 }
15
16 errorAnotaciones(){
17   for (let element of this.politicasReporte){
18     if (element.anoaciones==0){
```

```

19     return true
20   }
21   else{
22     return false
23   }
24 }
25 }
26

```

Código 2.19. Validación Formulario Reportes.

El Código 2.20 muestra la función que envía las políticas anotadas a ser descargadas hacia el *backend*.

```

1  generarReporte(){
2    if(this.politicasyReporte.length==0 || this.errorAnotaciones()){
3      this.errorCheckbox()
4      this.checkboxError=true;
5    }
6    else{
7      this.checkboxError=false;
8      this.reporte.politicasyReporte = this.politicasyReporte;
9      console.log(this.reporte)
10     this._politicaService.generarReporte(this.reporte).subscribe(
11       respuesta => {
12         let nota = '';
13         this.translate.stream("reporte.exito").subscribe((res: string)=>{
14           nota = res;
15         });
16         this.notificacion(nota, 'exito-snackbar')
17         this.archivo=respuesta.archivo;
18         console.log(this.archivo)
19       },
20       error => {
21         this.notificacion(error.error.mensaje? error.error.mensaje : 'ERROR','fracaso-snackbar')
22       }
23     )
24   }
25 }

```

Código 2.20. Envío de Políticas.

Las operaciones del *backend* están explicadas en tareas anteriores. Se presenta un ejemplo del resultado de la generación de reportes, se realizó el reporte de las políticas *Firefy* y *Nike* en ambos formatos. La Figura 2.20 muestra el formulario para elegir las políticas anotadas y el formato.

Reportes
 Generación de reportes de políticas de privacidad anotadas.

Selecciona el tipo de políticas anotadas
 Finalizadas

Filtrar política por nombre

#	Nombre Política	Usuario	Tipo de Usuario	Fecha de última Anotación	Número de Anotaciones	Seleccionar
1	Firefly	admin@gmail.com	Anotador	2023-01-18	7	<input checked="" type="checkbox"/>
2	3M	admin@gmail.com	Anotador	2023-01-19	4	<input checked="" type="checkbox"/>
3	Nike	admin@gmail.com	Consolidador	2023-01-21	6	<input type="checkbox"/>
4	Nike	admin@gmail.com	Anotador	2023-01-21	6	<input type="checkbox"/>
5	Firefly	david@gmail.com	Anotador	2023-01-19	7	<input type="checkbox"/>
6	Nike	david@gmail.com	Anotador	2023-01-21	6	<input type="checkbox"/>

Selecciona un formato para el reporte
 CSV

Generar Reporte

Figura 2.20. Formulario Reportes.

La Figura 2.21 muestra el fichero *Reportes* con tres archivos. Como fue mencionado los reportes en tipo YAML se genera uno por política, sucede lo contrario con el formato CSV que se genera todo en el mismo archivo.



Figura 2.21. Fichero de Reportes Generados.

La Figura 2.22 muestran los reportes que fueron generados.

```

1 policy_1:
2   contains_synthetic: false
3   policy_id: 21
4   policy_name: 'Firefly'
5   policy_type: TEST
6   segments:
7     > segments_1: ...
26    > segments_2: ...
41    > segments_3: ...
57    > segments_4: ...
74    > segments_5: ...
99    > segments_6: ...
124   > segments_7: ...
140

```

```

1 policy_2:
2   contains_synthetic: false
3   policy_id: 22
4   policy_name: 3M
5   policy_type: TEST
6   segments:
7     > segments_1: ...
24    > segments_2: ...
42    > segments_3: ...
61    > segments_4: ...
79

```

Figura 2.22. Reportes YAML Generados.

#	política	política_nombre	url	parrafo	parrafo_texto	anotacion	fecha_ano	permite	anotacion
2	21	Firefly	https://www.firefly.health/policies/privacy-policy	98	Welcome to the website (the "Website") for Firefly Health, Inc. ("Firefly Health") we	23	18/1/2023 17:10	VERDADERO	Welcome t
3	21	Firefly	https://www.firefly.health/policies/privacy-policy	99	This Privacy Policy describes the types of information we may collect from you or t	24	18/1/2023 17:10	FALSO	This Privac
4	21	Firefly	https://www.firefly.health/policies/privacy-policy	100	? This Privacy Policy does not cover the collection, use or disclosure of protected	25	18/1/2023 17:11	FALSO	This Privac
5	21	Firefly	https://www.firefly.health/policies/privacy-policy	102	Please read this Privacy Policy carefully to understand our policies and practices re	26	18/1/2023 17:11	VERDADERO	Please rea
6	21	Firefly	https://www.firefly.health/policies/privacy-policy	105	Information that you provide by filling in forms on our Services. This includes inform	27	18/1/2023 17:11	FALSO	Information
7	21	Firefly	https://www.firefly.health/policies/privacy-policy	105	Information that you provide by filling in forms on our Services. This includes inform	28	18/1/2023 17:11	VERDADERO	Details of t
8	21	Firefly	https://www.firefly.health/policies/privacy-policy	107	By voluntarily providing us with Personal Information, you are consenting to our us	29	18/1/2023 17:11	VERDADERO	By voluntar
9	22	3M	https://www.firefly.health/policies/privacy-policy	108	Los Sitios de Internet de 3M usan "cookies de sesión". Una cookie de sesión no li	37	19/1/2023 18:37	FALSO	Los Sitios
10	22	3M	https://www.firefly.health/policies/privacy-policy	110	Los Sitios de Internet de 3M también usan "cookies persistentes". Estas cookies	38	19/1/2023 18:38	VERDADERO	Los Sitios
11	22	3M	https://www.firefly.health/policies/privacy-policy	111	Algunos países requieren que otorgue su consentimiento para el uso de cookies c	39	19/1/2023 18:38	FALSO	Tenga en c
12	22	3M	https://www.firefly.health/policies/privacy-policy	113	En cualquier momento que visite un Sitio de Internet de 3M de alguno de estos paí:	40	19/1/2023 18:38	VERDADERO	En cualquier

Figura 2.23. Reporte CSV Generado.

2.3.4 IMPLEMENTACIÓN SPRINT 4 - INTERNACIONALIZACIÓN

Este *sprint* incluye la implementación de la internacionalización de la herramienta.

IMPLEMENTACION TAREA T-18 - INTERNACIONALIZACIÓN DE LA APLICACIÓN

La tarea T-18 consiste en la creación de funciones necesarias para la traducción de los menús del *frontend*, de tal forma que se pueda mostrar los textos de la aplicación en inglés o español según la preferencia del usuario.

El Código 2.21 muestra la modificación del *Header* de la herramienta para que el usuario pueda escoger el idioma de su preferencia.

```
1 <label>
2 <span>{{ 'menu_arbol.idioma' | translate }} </span>
3 </label> </label>
4 <select #langSelect (change)="escogerIdioma(langSelect.value)">
5 <option *ngFor="let lang of langs" [selected]="lang === ngSelect">
6 {{ lang }}
7 </option>
8 </select>
9 </label>
```

Código 2.21. HTML Header Principal.

Para realizar las traducciones se usó la biblioteca de internacionalización (i18n) para Angular ngx-translate. La Tabla 2.17 muestra la versión que se instaló.

Tabla 2.17. Versiones instaladas de ngx-translate.

Angular	@ngx-translate/core	@ngx-translate/http-loader
8	12.x+	4.x+

Después de seguir los pasos de instalación de la guía de *ngx-translate*, entonces se configura el archivo Typescript del *header* de la herramienta. Se usó como idioma por defecto el inglés. El Código 2.22 muestra la configuración del archivo mencionado. Si se quisiera agregar un nuevo idioma en el futuro basta con agregar a la lista *translate.addLangs*

```
constructor (
1 private _breakpointObserver: BreakpointObserver,
2 public _loginService: LoginService,
3 private _router :Router,
4 private translate: TranslateService,
5 )
6
7 {
8 this.translate.setDefaultLang('en');
9 this.translate.use('en');
```

```
this.translate.addLangs(['en', 'es'])
)
```

Código 2.22. Añadir Idiomas.

A continuación, se crea los archivos tipo JSON que realizan las traducciones. La Figura 2.24 muestra la ruta donde fueron creados los archivos y el Código 2.23 muestra un ejemplo de cómo se debe configurar el archivo de traducciones de inglés y español, respectivamente. Si se desea agregar otro idioma se tendrá que crear otro JSON con las siglas de la locación según la ISO 639-1.

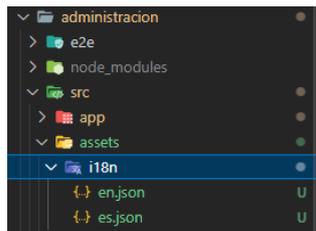


Figura 2.24. Archivos JSON.

```
1 "idiomas":{
2   "ingles": "English",
3   "espanol" : "Spanish"
4 },
5 "rol":{
6   "anotador" : "Annotator",
7   "admin" : "Administrator"
8 },
9 "botones":{
10  "guardar": "Save",
11  "cancelar" : "Cancel",
12  "siguiente" : "Next",
13  "extraer" : "Extract",
14 }
15 }
```

```
1 "idiomas":{
2   "ingles": "Ingles",
3   "espanol" : "Español"
4 },
5 "rol":{
6   "anotador" : "Anotador",
7   "admin" : "Administrator"
8 },
9 "botones":{
10  "guardar": "Guardar",
11  "cancelar" : "Cancelar",
12  "siguiente" : "Siguiente",
13  "extraer" : "Extraer",
14 }
15 }
```

Código 2.23. JSON de Traducciones.

Una vez que tengamos los archivos JSON completos con todos los pares: clave - valor se usa el servicio *TranslateService* para obtener los valores de traducción en los archivos Typescript, el Código 2.24 muestra un ejemplo de cómo hacerlo.

```

1 this.translate.stream("idiomas.espanol").subscribe((res: string)=>{
2     this.ngSelect = res;
3 });

```

Código 2.24. Traducciones en Archivos TS.

Y para obtener los valores de traducción en los archivos HTML se usó el pipe de `ngx-translate`. El Código 2.25 muestra un ejemplo de cómo hacerlo.

```

1 {{ 'menu_arbol.idioma' | translate }}

```

Código 2.25. Traducciones en Archivos HTML.

Las traducciones de los textos recuperados del *backend* están fuera del alcance del proyecto. Por este motivo, no se ha logrado realizar las traducciones del menú principal (*tree-view*). Sin embargo, el *Product Owner* del equipo ha establecido establecer el texto de los módulos a inglés. El Código 2.26 muestra la configuración en las migraciones de la aplicación para que se carguen los módulos en inglés.

```

1 op.bulk_insert(modulo_table,
2     #Inglés
3     [
4         ('id': 1, 'nombre': 'raiz', 'icono': '', 'path': None, 'padre_id': None),
5         ('id': 2, 'nombre': 'Administration', 'icono': 'build', 'path': None, 'padre_id': '1'),
6         ('id': 3, 'nombre': 'Consolidation', 'icono': 'book', 'path': '/paginas/consolidacion', 'padre_id': '1'),
7         ('id': 4, 'nombre': 'Annotation', 'icono': 'create', 'path': '/paginas/anotacion', 'padre_id': '1'),
8         ('id': 5, 'nombre': 'Users', 'icono': 'account_circle', 'path': '/paginas/administracion/usuarios', 'padre_id': '2'),
9         ('id': 6, 'nombre': 'Policies', 'icono': 'menu_book', 'path': None, 'padre_id': '2'),
10        ('id': 7, 'nombre': 'Treatments', 'icono': 'highlight', 'path': '/paginas/administracion/tratamientos', 'padre_id': '2'),
11        ('id': 9, 'nombre': 'Attributes', 'icono': 'format_list_bulleted', 'path': '/paginas/administracion/atributos', 'padre_id': '2'),
12        ('id': 10, 'nombre': 'Values', 'icono': 'format_list_numbered', 'path': '/paginas/administracion/valores', 'padre_id': '2'),
13        ('id': 11, 'nombre': 'Reports', 'icono': 'insert_drive_file', 'path': '/paginas/administracion/reportes', 'padre_id': '2'),
14        ('id': 12, 'nombre': 'Manual', 'icono': 'create', 'path': '/paginas/administracion/politicas/gestion', 'padre_id': '6'),
15        ('id': 13, 'nombre': 'Automatic', 'icono': 'note_add', 'path': '/paginas/administracion/politicas/insertar', 'padre_id': '6'),
16    ]
17 )

```

Código 2.26. Corrección Idioma de Módulos.

La Figura 2.25 muestra cómo queda la base de datos después de la configuración.

id	nombre	icono	path	padre_id
1	raiz		[NULL]	[NULL]
2	Administration	build	[NULL]	1
3	Consolidation	book	/paginas/cons	1
4	Annotation	create	/paginas/annot	1
5	Users	account_circle	/paginas/adm	2
6	Policies	menu_book	[NULL]	2
7	Treatments	highlight	/paginas/adm	2
9	Attributes	format_list_bul	/paginas/adm	2
10	Values	format_list_nur	/paginas/adm	2
11	Reports	insert_drive_file	/paginas/adm	2
12	Manual	create	/paginas/adm	6
13	Automatic	note_add	/paginas/adm	6

Figura 2.25. Módulos Corregidos.

La Figura 2.26 muestra cómo se despliegan los textos al iniciar la herramienta.

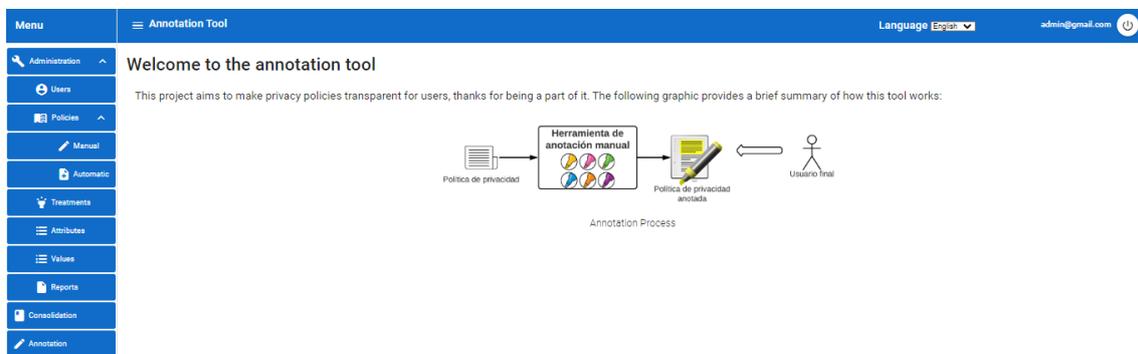


Figura 2.26. Interfaz Bienvenida Corregida.

2.4 VALIDACIÓN Y VERIFICACIÓN

Para la validación y verificación se ha realizado un proceso de comprobación y análisis que aseguran que la herramienta que se ha desarrollado está acorde a las especificaciones del *Product Owner*.

El objetivo es presentar la evidencia del funcionamiento de la herramienta de anotación. Este proceso se llevó a cabo mediante pruebas funcionales de la herramienta donde se valida que los criterios de aceptación de cada historia de usuario se cumplen. Para ello se realizó la extracción de texto de dos políticas de privacidad, anotación de las políticas y generación de reportes de las anotaciones.

2.4.1 PRUEBAS SPRINT 1 – DESPLIEGUE ENTORNO DE DESARROLLO

Contempla las historias de usuario no funcionales al desplegar la herramienta. En estas subsecciones nos centraremos en pruebas funcionales.

2.4.2 PRUEBAS SPRINT 2 – EXTRACCIÓN POLÍTICAS

El *sprint 2* corresponde a las historias de usuario: Políticas individuales (HU-05) y Políticas grupales (HU-06). Se ha validado los criterios de aceptación de las historias HU-05 y HU-06 en conjunto debido a que se ha implementado de tal forma que sea indiferente al ingresar una URL o al ingresar n URLs.

CRITERIOS DE ACEPTACIÓN HU-05 - POLÍTICAS INDIVIDUALES

Para la historia de usuario HU-05 los criterios de aceptación son:

1. Debe cumplir que el usuario ingresa una URL.
2. Debe cumplir que la herramienta extrae el texto de la política de privacidad automáticamente.
3. Debe cumplir que el usuario ingrese nombre y fecha de la política.
4. Debe cumplir que el usuario pueda editar el texto extraído.
5. Debe cumplir que se pueda guardar la política.

CRITERIOS DE ACEPTACIÓN DE HU-06 - POLÍTICAS GRUPALES

Para la historia de usuario HU-06 los criterios de aceptación son:

1. Debe cumplir que la herramienta pida una ruta de un fichero.
2. Debe cumplir que se extraiga el texto de un conjunto de políticas de privacidad automáticamente a la vez.
3. Debe cumplir que el usuario ingrese nombre y fecha de las políticas.
4. Debe cumplir que el usuario pueda editar los textos extraídos.
5. Debe cumplir que se pueda guardar las políticas.

PRUEBAS FUNCIONALES

El usuario debe crear un archivo en formato TXT donde ingresa todas las URLs que quiera extraer. La Figura 2.27 muestra la creación del archivo en formato TXT con dos URLs.



Figura 2.27. Archivo de Texto para Ingresar.

Se debe ingresar a la herramienta como administrador. En el módulo *Administrator* → *Policies* → *Automatic*, la herramienta pedirá la dirección de un fichero. La Figura 2.28 muestra como el usuario debe escoger su archivo de texto.

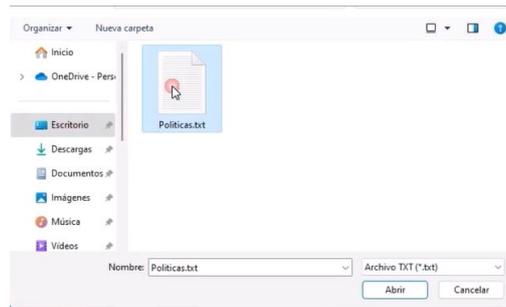


Figura 2.28. Selección de Archivo de Texto.

La herramienta validará el número de URLs del archivo. La Figura 2.29 muestra el mensaje enviado al usuario con el número de URLs.

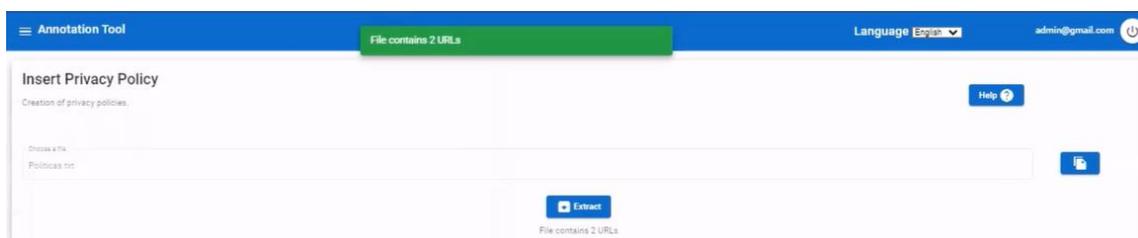


Figura 2.29. Validación de Archivo de Texto.

Al presionar en el botón *Extract* empezará el proceso de descarga del texto de la política de privacidad. Por cada URL se debe esperar máximo 30 segundos, que es el tiempo máximo que le toma al microservicio realizar la extracción. La Figura 2.30 muestra los mensajes que se generan al presionar el botón.

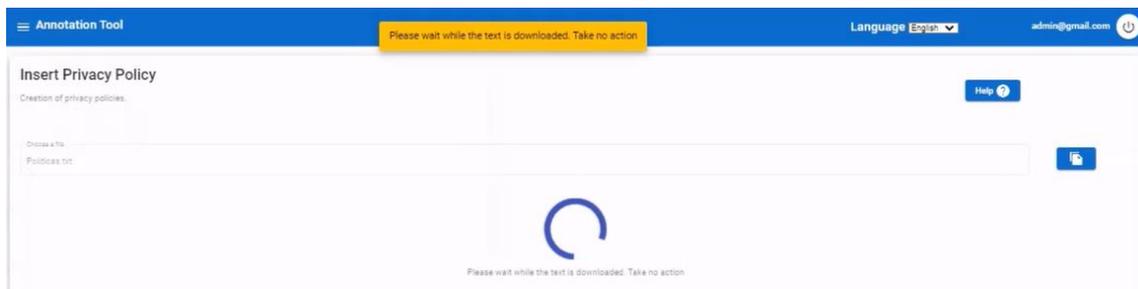


Figura 2.30. Proceso de Extracción de Texto.

Una vez extraído, se muestra un dialogo con toda la información recopilada de la política de privacidad. El usuario está en la capacidad de editar cualquier campo excepto la URL. La Figura 2.31 muestra el diálogo generado automáticamente.

The screenshot shows a web form titled "Insert privacy policy". It contains the following fields:

- Policy name ***: An empty text input field.
- URL where the policy was obtained ***: A text input field containing the URL `https://municipiodemejia.gob.ec/uploads/politicas/politica-privacidad-app.pdf`.
- Date the policy was obtained ***: A date input field containing `2023-02-24`.
- Privacy policy text**: A large text area containing the following text:


```
1. IDENTIFICACIÓN DEL RESPONSABLE DEL TRATAMIENTO.
EL Municipio de Mejía, es la entidad responsable del tratamiento de los datos facilitados por los clientes de la aplicación (en adelante, el/los Usuario/s).
2. FINALIDAD DEL TRATAMIENTO DE DATOS.
```

At the bottom of the form, there are three buttons: a red "Cancel" button, a red "Save" button, and a blue "Next" button.

Figura 2.31. Resultado de Extracción.

El usuario está en la capacidad de escoger si cancela la operación, guarda la política o avanza a la siguiente URL. La Figura 2.32 muestra como el usuario llena la información y guarda la política.

This screenshot shows the same "Insert privacy policy" form as in Figure 2.31, but with a confirmation dialog box overlaid on top. The dialog box is titled "Confirmation" and contains the text "Are you sure you want to save this policy?". It has three buttons: "No", "Yes", and "Cancel". A mouse cursor is pointing at the "Yes" button. The background form is dimmed.

Figura 2.32. Formulario Políticas.

Al guardar la política, automáticamente la herramienta empezará con el mismo proceso, pero con la siguiente URL. La Figura 2.33 muestra el mismo diálogo, pero con la segunda URL.

Figura 2.33. Extracción Siguiendo URL.

Una vez guardadas las políticas, se puede ver el resultado en el módulo *Administrator* → *Policies* → *Manual*. La figura 2.34 muestra las políticas ingresadas.

#	Name	URL	Date	Edit	Delete	Assign
1	Prueba1	https://municipiodemejia.gob.ec/uploads/politicas/politica-privacidad-app.pdf	2023-02-24			
2	Prueba2	https://explore.zoom.us/es/privacy/?preview_id=7880&preview_nonce=12ba08dbd56&preview=true	2023-02-24			

Figura 2.34. Políticas de Privacidad Ingresadas.

ANÁLISIS DE ACEPTACIÓN

En la Tabla 2.18 se muestra si las historias de usuario fueron terminadas, si cumplen con los criterios de aceptación y si fueron aceptadas por el *Product Owner*.

Tabla 2.18. Análisis de Aceptación HU-05, HU-06.

Código	Nombre Historia de Usuario	Terminada	Criterio de aceptación	Aceptación
HU-05	Políticas individuales	Sí	Debe cumplir que el usuario ingresa una URL.	Sí
			Debe cumplir que la herramienta extrae el texto de la política de privacidad automáticamente.	Sí

			Debe cumplir que el usuario ingrese nombre y fecha de la política.	Sí
			Debe cumplir que el usuario pueda editar el texto extraído.	Sí
			Debe cumplir que se pueda guardar la política.	Sí
HU-06	Políticas Grupales	Sí	Debe cumplir que la herramienta pida una ruta de un fichero.	Sí
			Debe cumplir que se extraiga el texto de un conjunto de políticas de privacidad automáticamente a la vez.	Sí
			Debe cumplir que el usuario ingrese nombre y fecha de las políticas.	Sí
			Debe cumplir que el usuario pueda editar los textos extraídos.	Sí
			Debe cumplir que se pueda guardar las políticas.	Sí

2.4.3 PRUEBAS SPRINT 3 – REPORTES

EL *sprint* 3 corresponde a la historia de usuario Reportes (HU-07).

CRITERIOS DE ACEPTACIÓN HU-07 - REPORTES

Para la historia de usuario HU-07 los criterios de aceptación son:

1. Debe cumplir que el usuario elija si se muestra políticas anotadas finalizadas o no finalizadas.
2. Debe cumplir que la herramienta muestre una tabla con políticas anotadas.
3. Debe cumplir que el usuario pueda escoger una o muchas políticas anotadas.
4. Debe cumplir que el usuario pueda escoger un formato para el reporte.
5. Debe cumplir que la herramienta genere el reporte de las políticas anotadas en el formato específico.

PRUEBAS FUNCIONALES

Continuando con el ejemplo anterior, se procede a anotar las políticas ingresadas para generar el reporte. La Figura 2.35 muestra 3 anotaciones realizadas a la política *Prueba1*.

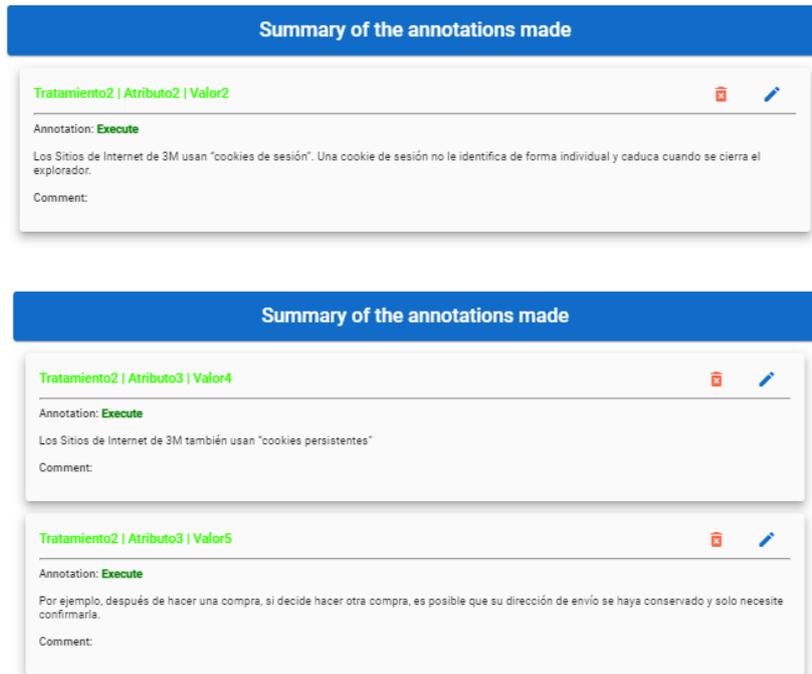


Figura 2.35. Anotaciones Prueba1.

. La Figura 2.36 muestra 2 anotaciones realizadas a la política *Prueba2*.

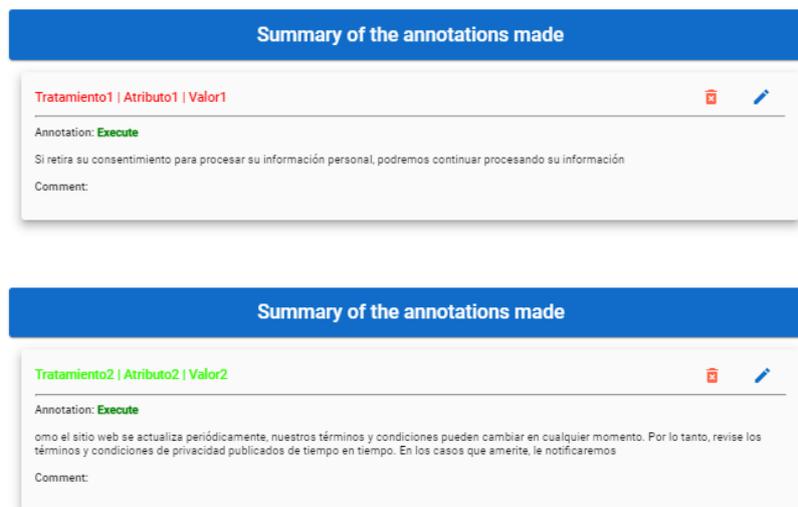


Figura 2.36. Anotaciones Prueba2.

Una vez que se ha realizado al menos 1 anotación ya es posible generar reportes en el módulo *Administrator* → *Reports*. En este caso en particular aún no se han finalizado de anotar las políticas completamente, es por eso por lo que cuando el usuario escoja la opción política anotadas finalizadas no se debe desplegar ninguna política. La figura 2.37 muestra el mensaje que muestra la herramienta.

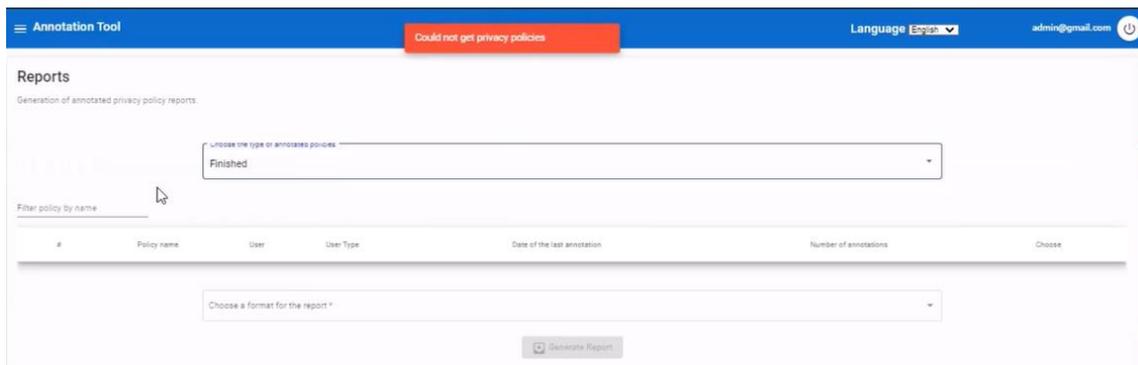


Figura 2.37. Herramienta sin Anotaciones.

En nuestro caso disponemos de políticas anotadas no finalizadas, al escoger dicha opción, se desplegará la tabla con las anotaciones de cada política. La tabla contiene el número de anotaciones realizadas por un usuario, el tipo de usuario (anotador o consolidador) y la fecha en la que se realizó la última anotación. La figura 2.38 muestra el resultado.

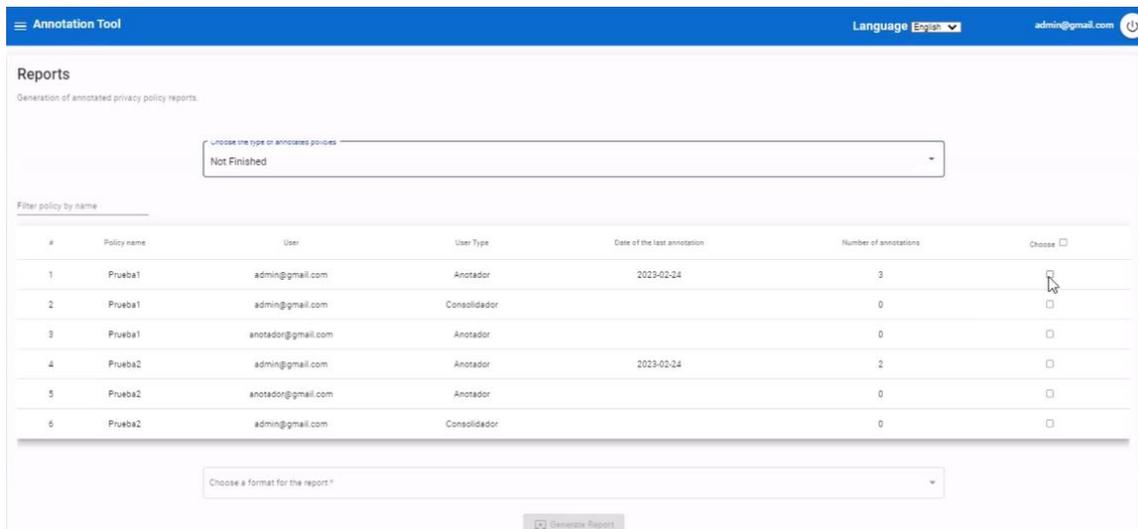


Figura 2.38. Herramienta con Anotaciones.

Se puede escoger una o muchas políticas anotadas siempre y cuando tengan un número de anotaciones mayor a 0. La Figura 2.39 muestra que, al llenar el formulario, se desbloquea el botón para generar el reporte.

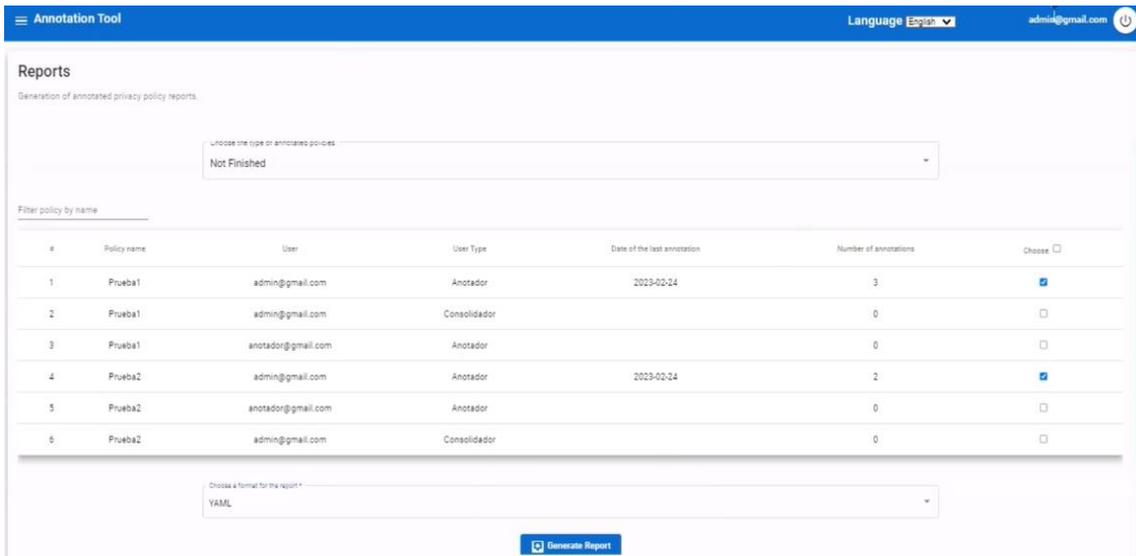


Figura 2.39. Formulario Reportes.

Se generará los reportes en ambos formatos en YAML y en CSV. La Figura 2.40 muestra el mensaje de la herramienta al dar *click* en el botón *Generate Report*.

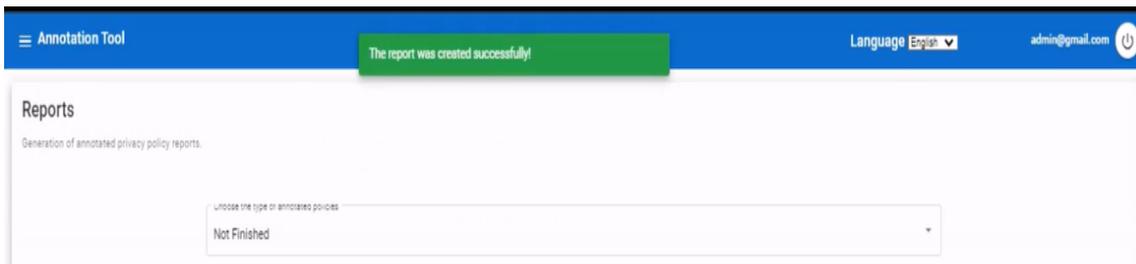


Figura 2.40. Generación de Reporte.

Los reportes se guardan en la ruta: *Backend* → *Reportes*. En el caso del formato YAML se genera 1 reporte por cada política. La Figura 2.41 muestra los reportes generados por la herramienta.

 Reporte-2023-02-24-15-30-35.csv	24/2/2023 15:30	Archivo de valores...	5 KB
 Reporte-Prueba1-2023-02-24-15-30-40.yml	24/2/2023 15:30	Archivo de origen ...	4 KB
 Reporte-Prueba2-2023-02-24-15-30-40.yml	24/2/2023 15:30	Archivo de origen ...	3 KB

Figura 2.41. Reportes Generados.

Finalmente, las Figura 2.42 muestra el contenido del reporte en CSV y la Figura 2.43 muestra el reporte en formato YAML.

política	nombre	url	parrafo_id	parrafo_texto
Prueba1		https://municipiodemejia.gob.ec/uploads/politicas/politica-privacidad-app.pdf	1	Los Sitios de Internet de 3M usa
Prueba1		https://municipiodemejia.gob.ec/uploads/politicas/politica-privacidad-app.pdf	3	Los Sitios de Internet de 3M tam
Prueba1		https://municipiodemejia.gob.ec/uploads/politicas/politica-privacidad-app.pdf	3	Los Sitios de Internet de 3M tam
Prueba2		https://explore.zoom.us/es/privacy/?preview_id=7880&preview_nonce=12ba08dbd5&prev	8	Si retira su consentimiento para
Prueba2		https://explore.zoom.us/es/privacy/?preview_id=7880&preview_nonce=12ba08dbd5&prev	10	Como el sitio web se actualiza pi

Figura 2.42. Reporte CSV.

```

1 policy_1:
2   contains_synthetic: false
3   policy_id: 1
4   policy_name: Prueba1
5   policy_type: TEST
6   segments:
7     > segments_1: ...
24    > segments_2: ...
49

```

Figura 2.43. Reporte YAML.

ANÁLISIS DE ACEPTACIÓN

En la Tabla 2.19 se muestra si las historias de usuario fueron terminadas, si cumplen con los criterios de aceptación y si fueron aceptadas por el *Product Owner*.

Tabla 2.19. Análisis de Aceptación HU-07.

Código	Nombre Historia de Usuario	Terminada	Criterio de aceptación	Aceptación
HU-07	Reportes	Sí	Debe cumplir que el usuario elija si se muestra políticas anotadas finalizadas o no finalizadas.	Sí
			Debe cumplir que la herramienta muestre una tabla con políticas anotadas.	Sí
			Debe cumplir que el usuario pueda escoger una o muchas políticas anotadas.	Sí
			Debe cumplir que el usuario pueda escoger un formato para el reporte.	Sí
			Debe cumplir que la herramienta genere el reporte de las políticas anotadas en el formato específico.	Sí

2.4.4 PRUEBAS SPRINT 4 – INTERNACIONALIZACIÓN

EL *sprint* 4 corresponde a la historia de usuario Internacionalización de la aplicación (HU-04).

CRITERIOS DE ACEPTACIÓN HU-04 - INTERNACIONALIZACIÓN

Para la historia de usuario HU-04 los criterios de aceptación son:

1. Debe cumplir que la herramienta se despliegue inicialmente en idioma inglés.
2. Debe cumplir que el usuario este en la capacidad de elegir el idioma en que la herramienta despliegue todos sus módulos en cualquier momento.

PRUEBAS FUNCIONALES

La Figura 2.44 muestra la pantalla de bienvenida al iniciar la herramienta, se puede notar que el texto se ha desplegado en inglés.

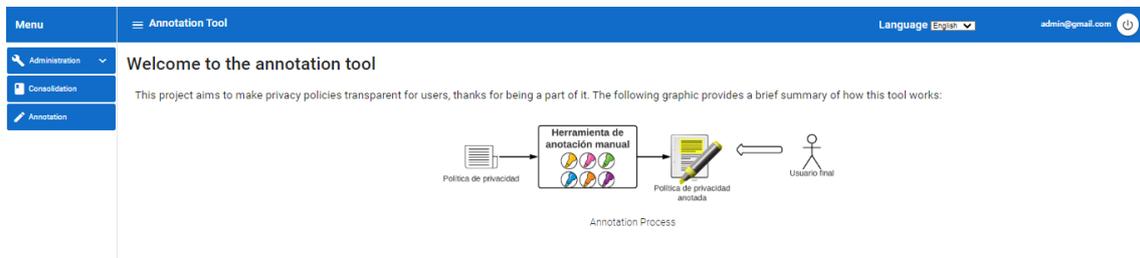


Figura 2.44. Interfaz de Bienvenida.

En la parte superior derecha de la pantalla el usuario tiene una opción donde podrá cambiar el idioma en cualquier momento. Las Figuras 2.45 y 2.46 muestran el cambio de inglés a español en el módulo *Users*.

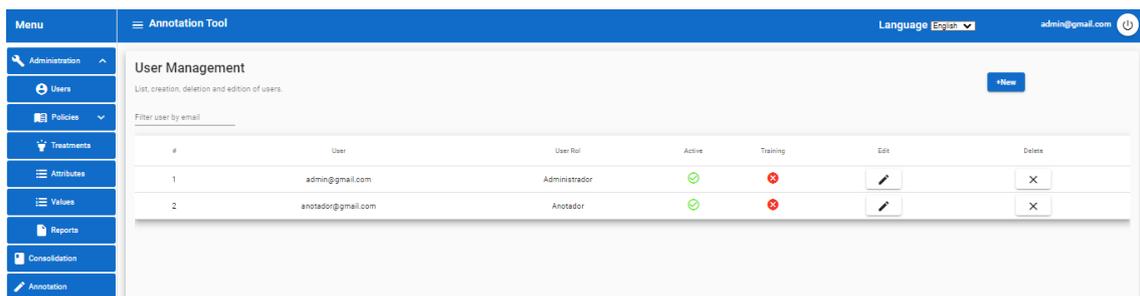


Figura 2.45. Interfaz Inglés.

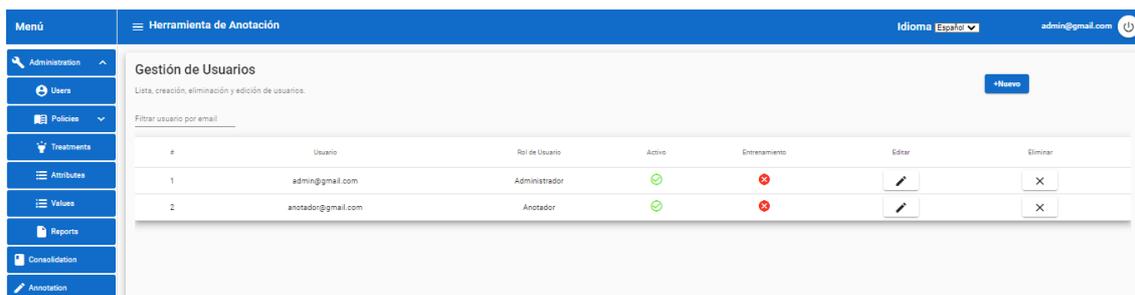


Figura 2.46. Interfaz Español

ANÁLISIS DE ACEPTACIÓN

En la Tabla 2.20 se muestra si las historias de usuario fueron terminadas, si cumplen con los criterios de aceptación y si fueron aceptadas por el *Product Owner*.

Tabla 2.20. Análisis de Aceptación HU-04.

Código	Nombre Historia de Usuario	Terminada	Criterio de aceptación	Aceptación
HU-04	Internacionalización de la aplicación	Sí	Debe cumplir que la herramienta se despliegue inicialmente en idioma inglés.	Sí
			Debe cumplir que el usuario este en la capacidad de elegir el idioma en que la herramienta despliegue todos sus módulos en cualquier momento.	Sí

2.4.5 RESUMEN

En la Tabla 2.21 se resume todas las historias de usuario con sus criterios de aceptación que fueron analizadas.

Tabla 2.21. Resumen de Aceptación.

Código	Nombre Historia de Usuario	Terminada	Criterio de aceptación	Aceptación
HU-04	Internacionalización de la aplicación	Sí	Debe cumplir que la herramienta se despliegue inicialmente en idioma inglés.	Sí

			Debe cumplir que el usuario este en la capacidad de elegir el idioma en que la herramienta despliegue todos sus módulos en cualquier momento.	Sí
HU-05	Políticas individuales	Sí	Debe cumplir que el usuario ingresa una URL.	Sí
			Debe cumplir que la herramienta extrae el texto de la política de privacidad automáticamente.	Sí
			Debe cumplir que el usuario ingrese nombre y fecha de la política.	Sí
			Debe cumplir que el usuario pueda editar el texto extraído.	Sí
			Debe cumplir que se pueda guardar la política.	Sí
HU-06	Políticas Grupales	Sí	Debe cumplir que la herramienta pida una ruta de un fichero.	Sí
			Debe cumplir que se extraiga el texto de un conjunto de políticas de privacidad automáticamente a la vez.	Sí
			Debe cumplir que el usuario ingrese nombre y fecha de las políticas.	Sí
			Debe cumplir que el usuario pueda editar los textos extraídos.	Sí
			Debe cumplir que se pueda guardar las políticas.	Sí

HU-07	Reportes	Sí	Debe cumplir que el usuario elija si se muestra políticas anotadas finalizadas o no finalizadas.	Sí
			Debe cumplir que la herramienta muestre una tabla con políticas anotadas.	Sí
			Debe cumplir que el usuario pueda escoger una o muchas políticas anotadas.	Sí
			Debe cumplir que el usuario pueda escoger un formato para el reporte.	Sí
			Debe cumplir que la herramienta genere el reporte de las políticas anotadas en el formato específico.	Sí

3 CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presenta las conclusiones y recomendaciones obtenidas en el desarrollo del presente trabajo de integración curricular

3.1 CONCLUSIONES

- Al finalizar el presente trabajo de integración curricular se dispone de la versión actualizada de la herramienta web AnooTool-Clip, con un conjunto de nuevas funcionalidades que permite mejorar la experiencia de los usuarios al realizar anotaciones de los tratamientos de datos personales que se informan a través de políticas de privacidad en español.
- Los fundamentos teóricos estudiados para el desarrollo de este proyecto sobre privacidad y protección de datos personales permitieron entender la necesidad e importancia de la herramienta AnooTool-Clip. Esta herramienta sirve de soporte a las técnicas para la evaluación automatizada de cumplimiento de políticas de privacidad en los sistemas de software.
- Los fundamentos teóricos estudiados sobre las tecnologías necesarias para el desarrollo de las nuevas funcionalidades que son los lenguajes de programación Python y TypeScript y los frameworks Flask y Angular han servido como

refuerzos y/o complementos a la formación académica como ingeniero en Tecnologías de la Información.

- Al usar Scrum se ha adoptado una estrategia de desarrollo iterativo e incremental, donde el producto va en continua evolución y mejora. Scrum permitió una correcta organización y planificación de las tareas propuestas.
- Aunque la herramienta web AnooTool-Clip para anotación es muy útil, aún se requieren funcionalidades nuevas. En este sentido, durante el desarrollo de esta, se ha mantenido la lógica y la esencia de la primera versión de la herramienta, con buenas prácticas de programación, como la legibilidad del código, reutilización de funciones, organización modular del código y comentarios indicativos. Esto con el objetivo de poder continuar mejorando y poder escalar hacia nuevas funcionalidades en el futuro.
- En la sección de validación y verificación se comprobó mediante pruebas de funcionamiento que la herramienta cumple con los requerimientos funcionales y no funcionales recolectados en la sección de análisis.

3.2 RECOMENDACIONES

- La herramienta de anotación debería tener incluido aún más idiomas que los que han sido incorporados en el presente trabajo. Lo que resultaría relativamente sencillo después de que se ha dejado un escenario listo para escalar. En la sección de implementación del *sprint* 4 se explica este proceso.
- En el presente trabajo se ha contemplado las traducciones del *frontend* de la herramienta de anotación, se sugiere que en futuros trabajos se tenga en cuenta las traducciones de los datos obtenidos desde el *backend* y además de incluir a la herramienta de visualización.
- El nuevo módulo para la generación de reportes debe ser revisado junto al equipo que se encarga de *Machine Learning*, esto con el objetivo de ajustar el estilo del reporte a sus necesidades. En el presente trabajo no se lo realizó debido a problemas de tiempo y alcance.
- La extracción automática de texto de políticas de privacidad mediante una URL contiene limitaciones en algunas URLs, esto es debido al Microservicio 3 de Julio Cesar Miranda. El autor sugiere que podría ser mejorado, abordando nuevas casuísticas.
- Se recomienda el uso de contenedores Docker para el despliegue de la herramienta, ya que facilita la ejecución independientemente del entorno físico.

4 REFERENCIAS

- [1] F. MIÑO, DESARROLLO DE UNA HERRAMIENTA WEB PARA LA ANOTACIÓN DE TRATAMIENTOS DE DATOS PERSONALES EN POLÍTICAS DE PRIVACIDAD EN ESPAÑOL, Quito: Escuela Politecnica Nacional, 2021.
- [2] Asamblea Nacional del Ecuador, «Ley Orgánica de Protección de Datos Personales,» 2021.
- [3] Oxford Dictionary, «Oxford Dictionary,» [En línea]. Available: <https://www.oxfordlearnersdictionaries.com/us/>. [Último acceso: Noviembre 2022].
- [4] Real Academia de la Lengua Española, Real Academia Española., 23, 2014.
- [5] G. R. Martínez, «Los desafíos de la privacidad y proteccion de datos en el 2021,» 23 02 2021. [En línea]. Available: <https://blog.alestra.com.mx/los-desafios-de-la-privacidad-y-proteccion-de-datos-en-el-2021>. [Último acceso: Noviembre 2022].
- [6] Union Europea, Reglamento General de Protección de Datos”, PARLAMENTO EUROPEO Y DEL CONSEJO, 2016.
- [7] OEA, «Principios Actualizados sobre la Privacidad y la Protección de Datos Personales,» 2021.
- [8] C. Troncoso, «Privacy & Online Rights Knowledge Area,» 2019.
- [9] A. Babatzikis, «Estas son las mejores prácticas en materia de política de privacidad y protección de datos,» qualifio, 2020. [En línea]. Available: <https://qualifio.com/es/blog/practicas-politica-privacidad-proteccion-datos/>. [Último acceso: Diciembre 2022].
- [10] Angular, «Angular - Introduction to Angular concepts,» [En línea]. Available: <https://angular.io/guide/architecture>. [Último acceso: Noviembre 2022].
- [11] Python, «What is Python? Executive Summary,» [En línea]. Available: <https://www.python.org/doc/essays/blurb/>. [Último acceso: Noviembre 2022].
- [12] Flask, «Application Factories — Flask Documentation,» [En línea]. Available: <https://flask.palletsprojects.com/en/1.1.x/patterns/appfactories/>. [Último acceso: Noviembre 2022].
- [13] J. Sutherland y K. Schwaber, «La Guía de Scrum,» 2020.

- [14] SCRUM, «Marco de Trabajo Scrum,» [En línea]. Available: <https://www.scrum.org/>. [Último acceso: Enero 2023].
- [15] J. C. MIRANDA, DESARROLLO DE MICROSERVICIOS PARA LA EXTRACCIÓN DE POLÍTICAS DE PRIVACIDAD DESDE ARTEFACTOS DE UNA APLICACIÓN MÓVIL ANDROID, Quito: Escuela Politécnica Nacional, 2022.
- [16] Y. FERNÁNDEZ, «XACATACA BASICS,» 15 Julio 2021. [En línea]. Available: <https://www.xataka.com>. [Último acceso: 22 Diciembre 2022].
- [17] Angular Material, «Material Design,» [En línea]. Available: <https://material.angular.io/>. [Último acceso: Febrero 2023].
- [18] PrivacyPolicies.com, «Privacy Policies,» 22 Julio 2022. [En línea]. Available: <https://www.privacypolicies.com/es/blog/politicas-privacidad-obligatorias-legalmente/>. [Último acceso: Noviembre 2022].
- [19] Chakray, «¿Qué son los microservicios? Características y ventajas,» [En línea]. Available: <https://www.chakray.com/es/que-son-los-microservicios-definicion-caracteristicas-y-ventajas-y-desventajas/>. [Último acceso: Diciembre 2022].
- [20] Figma, «Figma,» [En línea]. Available: <https://www.figma.com/>. [Último acceso: Diciembre 2022].
- [21] Electron, «Selenium and WebDriver,» 2021. [En línea]. Available: <https://www.electronjs.org/es/docs/latest/tutorial/using-selenium-and-webdriver#:~:text=ChromeDriver%20es%20un%20servidor%20independiente,equipos%20de%20Chromium%20y%20WebDriver..> [Último acceso: Diciembre 2022].
- [22] Comisión Europea, «Web oficial de la Union Europea,» [En línea]. Available: https://commission.europa.eu/law/law-topic/data-protection/reform/what-personal-data_es. [Último acceso: Diciembre 2022].
- [23] Universidad Andina Simon Bolivar, «Observatorio de Ciberderechos y Tecnosociedad,» 2021. [En línea]. Available: <https://www.uasb.edu.ec/ciberderechos/proteccion-de-datos/#:~:text=La%20protecci%C3%B3n%20de%20datos%20personales,directiva%2095%2F46%2FCE..> [Último acceso: Diciembre 2022].

5 ANEXOS

En esta sección se presenta los anexos que fueron necesarios para el trabajo:

ANEXO A: Historias de usuario.

ANEXO B: Product Backlog.

ANEXO C: Sprint Backlog.

ANEXO D: Interfaces de usuario (mockups) completas.

ANEXO E: Análisis de aceptación.

ANEXO F: Código fuente de la herramienta.

ANEXO G: Despliegue de la Herramienta de forma local.

ANEXO H: Demo de la aplicación.

ANEXO A

HISTORIAS DE USUARIO

Se presenta el listado completo de las historias de usuario:

Código	HU-00
Nombre	Nombre descriptivo
Descripción	Yo [<i>nombre solicitador</i>] como [<i>rol de la herramienta de anotación</i>] deseo [<i>funcionalidad</i>] para [<i>objetivo</i>]
Criterios de aceptación	<ul style="list-style-type: none">• Debe cumplir [<i>condición 1</i>]• Debe cumplir [<i>condición 2</i>]• ...• Debe cumplir [<i>condición n</i>]

Código	HU-01
Nombre	Adaptación de tecnologías
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que todo el conjunto de nuevas funcionalidades sea implementado con las mismas tecnologías con las que se desarrolló la primera versión de la herramienta para que exista compatibilidad de la aplicación y facilidad de desarrollo.
Criterios de aceptación	Debe cumplir que la PC de desarrollo tenga las mismas tecnologías usadas en la primera versión de la herramienta.

Código	HU-02
Nombre	Compatibilidad de actualizaciones
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que se resuelva las incompatibilidades debido a actualizaciones de bibliotecas usadas en versiones anteriores o debido a actualizaciones de la plataforma subyacente para evitar que las diferentes tecnologías de desarrollo, bibliotecas o paquetes presentes problemas de incongruencia en las nuevas versiones.
Criterios de aceptación	Debe cumplir que haya cero problemas de incongruencia entre las versiones de tecnologías, bibliotecas y/o paquetes a usar

Código	HU-03
Nombre	Despliegue de la herramienta de anotación
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta de anotación sea estudiada, desplegada y ejecutada, para que los desarrolladores estén familiarizados con todas las funcionalidades implementadas en la primera versión.
Criterios de aceptación	Debe cumplir que la herramienta sea correctamente manipulada en todos sus módulos.

Código	HU-04
Nombre	Internacionalización de la aplicación
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo adaptar a la herramienta para mostrar información en al menos dos idiomas: inglés y español, para presentar a usuarios que hablan diferentes idiomas, una aplicación visual y funcionalmente idéntica.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que la herramienta se despliegue inicialmente en idioma inglés. • Debe cumplir que el usuario este en la capacidad de elegir el idioma en que la herramienta despliegue todos sus módulos en cualquier momento.

Código	HU-05
Nombre	Extracción automática de políticas individuales
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta sea capaz de extraer automáticamente el texto de una política de privacidad, mediante una URL, para facilitar este proceso a los usuarios.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que el usuario ingresa una URL. • Debe cumplir que la herramienta extrae el texto de la política de privacidad automáticamente. • Debe cumplir que el usuario ingrese nombre y fecha de la política. • Debe cumplir que el usuario pueda editar el texto extraído. • Debe cumplir que se pueda guardar la política.

Código	HU-06
Nombre	Extracción automática de políticas grupales
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta extraiga automáticamente el texto de un conjunto de políticas de privacidad mediante la asignación de una ruta con dirección a un fichero donde están alojadas varias URLs, para optimizar tiempo al ingresar URLs una por una y facilitar el proceso a los usuarios.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que la herramienta pida una ruta de un fichero. • Debe cumplir que se extraiga el texto de un conjunto de políticas de privacidad automáticamente a la vez. • Debe cumplir que el usuario ingrese nombre y fecha de las políticas. • Debe cumplir que el usuario pueda editar los textos extraídos. • Debe cumplir que se pueda guardar las políticas.

Código	HU-07
Nombre	Reporte de anotaciones
Descripción	Yo Product Owner como administrador de la herramienta de anotación deseo que la herramienta sea capaz de crear reportes en un formato semiestructurado especificado por el usuario, para crear datasets.
Criterios de aceptación	<ul style="list-style-type: none"> • Debe cumplir que el usuario elija si se muestra políticas anotadas finalizadas o no finalizadas. • Debe cumplir que la herramienta muestre una tabla con políticas anotadas. • Debe cumplir que el usuario pueda escoger una o muchas políticas anotadas. • Debe cumplir que el usuario pueda escoger un formato para el reporte. • Debe cumplir que la herramienta genere el reporte de las políticas anotadas en el formato específico.

ANEXO B

PRODUCT BACKLOG

Se presenta el *Product Backlog* final del trabajo organizado por prioridad:

Código	Nombre historia de usuario	Código tarea	Nombre tarea	Descripción	Prioridad
HU-01	Adaptación de tecnologías	T-01	Adaptación de tecnologías	Instalación de las mismas tecnologías usadas para el desarrollo de la primera versión de la herramienta: Python, Flask, Angular.	Alta
HU-02	Compatibilidad de actualizaciones	T-02	Compatibilidad de actualizaciones	Instalación de paquetes y bibliotecas necesarias y resolución de incompatibilidades debido a actualizaciones.	Alta
HU-03	Despliegue de la herramienta de anotación	T-03	Despliegue de la herramienta de anotación	Despliegue y manipulación funcional de la primera versión de la herramienta.	Alta
HU-05 / HU-06	Políticas individuales / Políticas grupales	T-04	Servidor Ubuntu	Creación de un servidor Ubuntu	Alta
		T-05	Despliegue microservicio	Despliegue del microservicio 3 del trabajo “Desarrollo de microservicios para la extracción de políticas de privacidad desde artefactos de una aplicación móvil android” desarrollado por Julio Cesar Miranda, en el servidor Ubuntu.	Alta
		T-06	Automatización Servidor Linux	Creación de un Script de Linux para automatizar configuración del servidor y despliegue del microservicio.	Alta

		T-07	Tree-view	Actualización del menú principal de la herramienta de manera que se muestren los nuevos módulos a incorporar.	Media
		T-08	Frontend políticas	Creación de interfaz gráfica (Frontend) para insertar políticas automáticamente.	Media
		T-09	Lógica Frontend de políticas	Creación de las funciones necesarios en el frontend para que el usuario pueda insertar una ruta con dirección a un archivo en formato TXT que contiene un conjunto de URLs y enviarlo al backend. (requerimientos del microservicio 3)	Media
		T-10	Conexión Backend y Servidor Linux	Creación de Script en Python para conexión entre backend de la herramienta de anotación y el nuevo servidor Ubuntu y envió del archivo en formato TXT.	Media
		T-11	Modificación microservicio	Adaptación y modificación del microservicio 3 hacia las nuevas necesidades.	Media
		T-12	Validación formulario políticas	Generación de formulario en el frontend para insertar datos de la política o modificar el texto de esta. Creación de lógica necesaria en el backend para guardar la nueva política en la base de datos.	Media

HU-07	Reportes	T-13	Consulta SQL	Creación de consulta SQL usando SQLAlchemy, con los datos necesarios para realizar reportes de políticas anotadas.	Media
		T-14	Reportes en CSV	Creación de Script de Python que genere reportes de políticas anotadas en formato CSV.	Media
		T-15	Reportes en YAML	Creación de Script de Python que genere reportes de políticas anotadas en formato YAML.	Media
		T-16	Tabla de anotaciones	Generación de interfaz gráfica que muestre una tabla con las políticas anotadas.	Media
		T-17	Validación formulario reportes	Creación y validación de formulario para escoger el tipo de políticas, el formato del reporte y las políticas anotadas.	Media
HU-04	Internacionalización de la aplicación	T-18	Internacionalización de la aplicación	Creación de funciones necesarias para la traducción de textos entre inglés y español.	Baja

ANEXO C

SPRINT BACKLOG

A continuación, en las siguientes tablas se detalla las tareas a realizar por cada *Sprint*.

N° Sprint		1		
Nombre Sprint		Despliegue del entorno de desarrollo		
Objetivo Sprint		Ejecutar y estudiar la primera versión de la herramienta de anotación		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-01	Adaptación de tecnologías	T-01	Alta	1
HU-02	Compatibilidad de actualizaciones	T-02	Alta	2
HU-03	Despliegue de la herramienta de anotación	T-03	Alta	3

N° Sprint		2		
Nombre Sprint		Extracción de políticas		
Objetivo Sprint		Crear módulo de extracción automática de políticas individuales y grupales		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-05 / HU-06	Políticas Individuales / Políticas Grupales	T-04	Alta	1
		T-05	Alta	3
		T-06	Media	2
		T-07	Media	3
		T-08	Media	2
		T-09	Media	3
		T-10	Media	5
		T-11	Media	1
T-12	Media	5		

N° Sprint		3		
Nombre Sprint		Reporte de anotaciones		
Objetivo Sprint		Crear módulo de reportes		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-07	Reportes	T-13	Media	5
		T-14	Media	3
		T-15	Media	3
		T-16	Media	3
		T-17	Media	3

N° Sprint		4		
Nombre Sprint		Internacionalización		
Objetivo Sprint		Internacionalizar la herramienta de anotación		
Código H.U.	Nombre H.U.	Código Tarea	Prioridad	Estimación
HU-04	Internacionalización de la aplicación	T-18	Baja	5

ANEXO D

INTERFACES DE USUARIO (*MOCKUPS*)

Las interfaces de usuario o mockups completos se las puede encontrar en el siguiente link:

<https://www.figma.com/file/CgWOWTb3ti1RJnCJCNIYR6/Interfaces?node-id=0%3A1&t=SzkJx545cGQN0ctD-1>

Para obtener permisos de edición solicitar a: davidayd20@gmail.com

ANEXO E

ANÁLISIS DE ACEPTACIÓN

Se resume todas las historias de usuario con sus criterios de aceptación que fueron analizadas.

Código	Nombre Historia de Usuario	Terminada	Criterio de aceptación	Aceptación
HU-04	Internacionalización de la aplicación	Sí	Debe cumplir que la herramienta se despliegue inicialmente en idioma inglés.	Sí
			Debe cumplir que el usuario este en la capacidad de elegir el idioma en que la herramienta despliegue todos sus módulos en cualquier momento.	Sí
HU-05	Políticas individuales	Sí	Debe cumplir que el usuario ingresa una URL.	Sí
			Debe cumplir que la herramienta extrae el texto de la política de privacidad automáticamente.	Sí
			Debe cumplir que el usuario ingrese nombre y fecha de la política.	Sí
			Debe cumplir que el usuario pueda editar el texto extraído.	Sí
			Debe cumplir que se pueda guardar la política.	Sí
HU-06	Políticas Grupales	Sí	Debe cumplir que la herramienta pida una ruta de un fichero.	Sí
			Debe cumplir que se extraiga el texto de un conjunto de políticas de privacidad	Sí

			automáticamente a la vez.	
			Debe cumplir que el usuario ingrese nombre y fecha de las políticas.	Sí
			Debe cumplir que el usuario pueda editar los textos extraídos.	Sí
			Debe cumplir que se pueda guardar las políticas.	Sí
HU-07	Reportes	Sí	Debe cumplir que el usuario elija si se muestra políticas anotadas finalizadas o no finalizadas.	Sí
			Debe cumplir que la herramienta muestre una tabla con políticas anotadas.	Sí
			Debe cumplir que el usuario pueda escoger una o muchas políticas anotadas.	Sí
			Debe cumplir que el usuario pueda escoger un formato para el reporte.	Sí
			Debe cumplir que la herramienta genere el reporte de las políticas anotadas en el formato específico.	Sí

ANEXO F

CÓDIGO FUENTE

El código fuente se lo puede obtener en los siguientes repositorios:

Herramienta de Anotación:

https://github.com/DavidPro1997/Herramienta_Anotacion.git

Microservicio 3:

<https://github.com/DavidPro1997/microservice3.git>

ANEXO G

DESPLIEGUE DE LA HERRAMIENTA DE FORMA LOCAL

Microservice

Despliegue y configuración de microservicio

1. En una maquina Linux Ubuntu 18 abra un terminal
2. Descargue el proyecto

```
git clone https://github.com/DavidPro1997/microservice3.git
```

3. Ejecute el Script MicroService3.sh

```
sudo ./MicroService3.sh
```

4. En el archivo /etc/vsftpd.conf cambiar write_enable=YES

Herramienta

- En una carpeta local clone el repositorio donde este alojado el proyecto.

```
git clone https://github.com/DavidPro1997/Herramienta_Anotacion.git
```

Backend

1. Instale Python3 (≥ 3.6 $\leq 3.8.6$)
2. Vaya a la carpeta \Backend cree un ambiente virtual y actívelo.

```
py -m pip install virtualenv
py -m virtualenv de
.\de\Scripts\activate
```

3. En el ambiente virtual (de) instale las librerías necesarias.

```
py -m pip install -r requirements.txt
```

4. En MySQL cree una base de datos con cualquier nombre.
Ejemplo: dbHerramienta
5. En el archivo .env de la ruta: backend\.env actualice el string de conexión.

```
1 DB_CONN=mysql+pymysql://root@localhost/dbHerramienta
2 SECRET_KEY=Secret-that-needs-to-be-updated
3 APP_ENV=dev
4 FLASK_ENV=development
```

6. Ejecute las migraciones.

```
py manage.py db migrate --message 'Algun mensaje'  
py manage.py db upgrade
```

7. Vaya al archivo backend\app\main\service\politica_service.py y edite las credenciales del servidor Linux Ubuntu 18 donde este alojado el microservicio con unas propias

```
1 #Credenciales del servidor Ubuntu  
2 servidor = "192.168.1.128"  
3 usuario = "david"  
4 clave = "4652"
```

8. Guarde todo y corra el proyecto.

```
py manage.py run
```

Frontend

1. Instale Angular.

```
npm install -g @angular/cli
```

2. Desde la carpeta \administracion ejecute:

```
npm install  
ng serve
```

3. Desde la carpeta \visualizacion ejecute:

```
npm install  
ng serve --port=4201
```

4. Abra un navegador con la ruta <http://localhost:4200/>, ingrese las credenciales y compruebe su funcionamiento.
5. Abra un navegador con la ruta <http://localhost:4201/> y compruebe su funcionamiento.

Credenciales:

- Administrador:
 - email: admin@gmail.com
 - clave: admin
- Anotador:
 - email: anotador@gmail.com
 - clave: anotador

ANEXO H

DEMO DE LA APLICACIÓN

El demo de la aplicación se lo puede encontrar en el siguiente enlace:

https://drive.google.com/drive/folders/1qaQff_LQzfA4SyfduUG1WScEVHPeJcD?usp=sharing