



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Transcriptor: plataforma colaborativa para la transcripción de manuscritos

AUTORES: Christian Ezequiel Bracco - Lucas Ezequiel Cuevas

DIRECTOR/A: Diego Torres

CARRERA: Licenciatura en Sistemas

Resumen

La transcripción manual es la tarea en la que una persona lee un documento manuscrito y escribe en un entorno digital el texto que esta leyendo. Esta labor es un proceso largo el cual consume mucho tiempo para una persona. Sin embargo, la inteligencia de los transcriptores provee mejores resultados que las alternativas automáticas. Esta tesis introduce una plataforma de transcripción con enfoque colaborativo llamada Transcriptor, en la cual los miembros de una comunidad podrán cargar manuscritos digitalizados y colaborar en su transcripción, definiendo diferentes capas de transcripciones las cuales se representan usando tecnologías de web semántica y técnicas de gamificación. Asimismo, se realizaron distintos tipos evaluaciones para demostrar la efectividad y aceptación de la plataforma.

Palabras Clave

Anotaciones. Anotaciones Semánticas, Ciencia Ciudadana. Crowdsourcing, Gamificación. Manuscritos, Transcripción, Transcriptor

Conclusiones

La transcripción de manuscritos es una tarea ardua y meticulosa que se extiende mucho más allá de una simple transliteración del texto de una fuente a otra. Requiere de un proceso de definición, comprensión y organización de los contenidos que, debido a la falta de métodos automatizados confiables, le agrega complejidad y extiende sus tiempos de realización. Facilitar este trabajo unificando las principales tareas en una plataforma fue el puntapié inicial para el desarrollo de Transcriptor.

Trabajos Realizados

Se desarrolló una plataforma web para la transcripción colaborativa de manuscritos, facilitando herramientas para la transcripción textual y transcripción semántica, que permite a los colaboradores participar mediante transcripciones, proponer alternativas de transcripción y opinar en foros de discusión. En este contexto, la plataforma se integró con servicios de gamificación para registrar la actividad los usuarios y en base a esto cumplan objetivos para ir avanzando en una jerarquía de rangos. Se realizaron pruebas de usabilidad con voluntarios para detectar posibles mejoras y evaluar la efectividad de la plataforma frente a los objetivos propuestos.

Trabajos Futuros

La extensión de este trabajo involucra la incorporación de nuevas características a la plataforma, entre ellas, agregar edición de tiempo real, moderación de usuarios y herramientas de asistencia de reconocimiento automático. Desarrollar una aplicación móvil de transcripciones. Extender gamificación definiendo nuevos premios y desafíos. Extender las transcripciones semánticas, para conectar y exponer datos con otras ontologías. Definir nuevos formatos de exportación para los proyectos.

Fecha de la presentación: Febrero 2024



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Transcriptor: plataforma colaborativa para la transcripción de manuscritos

AUTORES: Christian Ezequiel Bracco - Lucas Ezequiel Cuevas

DIRECTOR/A: Diego Torres

CARRERA: Licenciatura en Sistemas

Resumen

La transcripción manual es la tarea en la que una persona lee un documento manuscrito y escribe en un entorno digital el texto que esta leyendo. Esta labor es un proceso largo el cual consume mucho tiempo para una persona. Sin embargo, la inteligencia de los transcriptores provee mejores resultados que las alternativas automáticas. Esta tesis introduce una plataforma de transcripción con enfoque colaborativo llamada Transcriptor, en la cual los miembros de una comunidad podrán cargar manuscritos digitalizados y colaborar en su transcripción, definiendo diferentes capas de transcripciones las cuales se representan usando tecnologías de web semántica y técnicas de gamificación. Asimismo, se realizaron distintos tipos evaluaciones para demostrar la efectividad y aceptación de la plataforma.

Palabras Clave

Anotaciones. Anotaciones Semánticas, Ciencia Ciudadana. Crowdsourcing, Gamificación. Manuscritos, Transcripción, Transcriptor

Conclusiones

La transcripción de manuscritos es una tarea ardua y meticulosa que se extiende mucho más allá de una simple transliteración del texto de una fuente a otra. Requiere de un proceso de definición, comprensión y organización de los contenidos que, debido a la falta de métodos automatizados confiables, le agrega complejidad y extiende sus tiempos de realización. Facilitar este trabajo unificando las principales tareas en una plataforma fue el puntapié inicial para el desarrollo de Transcriptor.

Trabajos Realizados

Se desarrolló una plataforma web para la transcripción colaborativa de manuscritos, facilitando herramientas para la transcripción textual y transcripción semántica, que permite a los colaboradores participar mediante transcripciones, proponer alternativas de transcripción y opinar en foros de discusión. En este contexto, la plataforma se integró con servicios de gamificación para registrar la actividad los usuarios y en base a esto cumplan objetivos para ir avanzando en una jerarquía de rangos. Se realizaron pruebas de usabilidad con voluntarios para detectar posibles mejoras y evaluar la efectividad de la plataforma frente a los objetivos propuestos.

Trabajos Futuros

La extensión de este trabajo involucra la incorporación de nuevas características a la plataforma, entre ellas, agregar edición de tiempo real, moderación de usuarios y herramientas de asistencia de reconocimiento automático. Desarrollar una aplicación móvil de transcripciones. Extender gamificación definiendo nuevos premios y desafíos. Extender las transcripciones semánticas, para conectar y exponer datos con otras ontologías. Definir nuevos formatos de exportación para los proyectos.

Fecha de la presentación: Febrero 2024

Resumen

La transcripción manual es la tarea en la que una persona lee un documento manuscrito y escribe en un entorno digital el texto que esta leyendo. Esta labor es un proceso largo el cual consume mucho tiempo para una persona. Sin embargo, la inteligencia de los transcriutores provee mejores resultados que las alternativas automáticas. Esta tesis introduce una plataforma de transcripción con enfoque colaborativo llamada Transcriptor, en la cual los miembros de una comunidad podrán cargar manuscritos digitalizados y colaborar en su transcripción, definiendo diferentes capas de transcripciones las cuales se representan usando tecnologías de web semántica y técnicas de gamificación. Asimismo, se realizaron distintos tipos evaluaciones para demostrar la efectividad y aceptación de la plataforma.

Agradecimientos

Expresamos nuestro más sincero agradecimiento a todos aquellos que contribuyeron para que este proyecto sea posible. Desde nuestros familiares, amigos y compañeros, cuyo apoyo incondicional fue fundamental a lo largo de todo este proceso.

De manera particular, agradecemos a Diego, cuya orientación y criterio fueron esenciales en el desarrollo de este trabajo. Su paciencia y pasión en su rol fueron un verdadero motor para nosotros.

A Gimena quien compartió su vasta experiencia y conocimiento en el ámbito de la transcripción y conservación de manuscritos históricos. Un dominio en el cual no nos encontramos interiorizados y que sin su guía y acompañamiento habría sido considerablemente más complejo de abordar.

Agradecemos también a todos los colaboradores que participaron en las pruebas, proporcionando valioso material que nos permitió evaluar la efectividad de Transcriptor.

Por último, queremos expresar nuestro profundo respeto y admiración por todos los profesionales y entusiastas que se dedican a la conservación y preservación de la información contenida en textos manuscritos. Su labor es fundamental para garantizar que el conocimiento perdure a lo largo del tiempo y continúe enriqueciendo nuestras vidas.

Los autores.

A mis padres, Edith y Flavio, mis hermanos, abuelos y familia, por apoyarme incondicionalmente siempre en cada paso y decisión de mi vida. Gran parte en la realización de objetivo es gracias a ustedes.

A Flor, mi novia, por elegir acompañarme en este camino y estar ahí en cada momento, celebrando cada pequeño logro y siendo mi refugio en momentos difíciles.

A mis compañeros y amigos de facultad, en quienes siempre encontré un gesto, un consejo, una risa o un mate que me alentaron durante este tiempo.

A Lucas, gran amigo, compañero y colega con quien tengo el placer de haber compartido toda una carrera y seguir trabajando juntos en nuevos proyectos.

Christian

No quiero dejar pasar la oportunidad de agradecer a esas personas que fueron pilares indispensables en este camino: A mis abuelos, padres y familia que siempre me acompañaron y apoyaron incondicionalmente.

A mis amigos de la facultad, con quienes compartimos grandes momentos, gracias a ustedes pude disfrutar aún más este recorrido, con quienes también pudimos crecer juntos de manera profesional.

A Christian, un gran amigo y compañero que me dejó esta carrera, con quien tuve el placer y privilegio de cursar y seguir trabajando en nuevos proyectos.

Lucas

Índice general

1. Introducción	6
1.1. Motivación	6
1.2. Objetivo	7
1.3. Objetivos específicos	7
1.4. Estructura del Documento	8
I Marco conceptual	9
2. Transcripción	10
2.1. Transcripción	10
2.2. Modos de transcripción	10
2.3. Tecnologías de transcripción de textos	11
2.4. Transcripción aplicada a la ciencia ciudadana	13
3. Web Semántica	15
3.1. Definición	15
3.2. Componentes de la web semántica.	15
3.2.1. Ontologías	15
3.2.2. RDF y RDFS	15
3.2.3. OWL	16
3.2.4. Linked Data	16
3.2.5. JSON-LD	17
4. Gamificación	18
4.1. Definición y beneficios	18
5. Trabajo Relacionado	20
5.1. FromThePage	20
5.2. Zooniverse	22
5.2.1. Proyectos de zooniverse relacionados al mundo de la transcripción	23
5.2.2. Transcribe Cooper	24
5.2.3. Workflow de Transcribe Cooper	24
5.3. El Proyecto Scribe	27
5.3.1. Características de los proyectos Scribe	28
5.3.2. Configuración de un proyecto Scribe	29
5.4. Anotación semántica de textos: Recogito	33
5.5. Conclusiones	35

II Diseño e Implementación	36
6. Estrategia general	37
6.1. Introducción	37
6.2. Características	38
6.2.1. Usuarios	38
6.2.2. Transcripciones	38
6.2.3. Capas de Transcripción	38
6.2.4. Capas de transcripción semántica	39
6.2.5. Marcas	40
6.2.6. Método de transcripción	40
6.2.7. Gamificación	40
6.2.8. Votación de Marcas	41
7. Arquitectura e Implementación	42
7.1. Introducción	42
7.2. FromThePage	42
7.3. Arquitectura General	43
7.4. Aplicación Core (transcriptor-backend)	44
7.4.1. Tecnologías Utilizadas/Implementación	44
7.4.2. Modelo	45
7.4.3. Modelo de interacción	47
7.4.4. API (transcriptor-api)	48
7.4.5. Persistencia	59
7.4.6. Gamificación	61
7.4.7. Transcripción semántica	64
7.4.8. Internacionalización en Servidor (Server-Side i18n)	70
7.4.9. Despliegue e Instalación	73
7.5. Frontend	76
7.5.1. Tecnologías Utilizadas	76
7.5.2. Tipos de módulos	76
7.5.3. Cliente HTTP	77
7.5.4. Secciones	79
7.5.5. Elementos de transcripción y generación de texto	88
7.5.6. Estrategias de transcripción	90
7.5.7. Competencia de transcripciones	98
7.5.8. Control de versiones	101
7.5.9. Definición de capas semánticas	103
7.5.10 Método de transcripción: Transcripción semántica	107
7.5.11 Internacionalización en Cliente (Client-Side i18n)	119
7.5.12 Despliegue e Instalación	121
8. Evaluaciones y Presentaciones	124
8.1. Evaluación	124
8.1.1. Objetivo	124
8.1.2. Evaluación de usabilidad con usuarios	124
8.1.3. Resultados esperados	125
8.1.4. Evaluación de usabilidad mediante recorrido cognitivo	131
8.2. Presentaciones	135
8.2.1. II Congreso de Ciencia Abierta y Ciudadana en Argentina (CIACIAR)	135

8.2.2. III Congreso Internacional de la Asociación Argentina de Humanidades Digitales (AAHD)	135
8.2.3. #NoviembreHD	137
8.2.4. III Congreso de Ciencia Abierta y Ciudadana en Argentina (CIACIAR)	137
8.2.5. HCI International 2023	138
9. Conclusión y trabajo futuro	139
9.1. Trabajo futuro	140
9.1.1. Mejoras en la herramienta de transcripción	140
9.1.2. Aplicación Móvil	141
9.1.3. Mejoras en la herramienta de transcripción semántica	141
9.1.4. Extender gamificación	141
Bibliografía	142
III Anexo	144
A. Documento presentado en HCI International 2023	145
B. Propuesta Evaluación de usuarios	165
C. Propuesta Recorrido cognitivo	173

Capítulo 1

Introducción

1.1. Motivación

La digitalización de documentos no es una actividad nueva para las bibliotecas e instituciones de patrimonio cultural. De hecho, se ha convertido en una herramienta fundamental para preservar, brindar y extender el acceso a colecciones de archivos que incluyen libros raros, manuscritos y fotografías antiguas [1] los cuales sufren de deterioro con el paso del tiempo. A su vez, el valor potencial de la investigación de las colecciones digitalizadas tampoco es un fenómeno nuevo. Sin embargo, la conversión de imágenes de contenido en datos legibles por máquina que pueden buscarse, ordenarse y manipularse de otra manera abre un nuevo abanico de dificultades y problemas a resolver.

En la actualidad la conversión de estos datos almacenados generalmente en imágenes estáticas se podría automatizar con varias herramientas que logran obtener texto editable en pocos segundos. Estos métodos avanzaron mucho al correr de los años volviéndose cada vez más fiables e inteligentes [2], pero con resultados sobre texto manuscrito aún poco favorables. Esto se debe a que no todos escriben de manera uniforme ni de la misma forma. Adicionalmente, la necesidad de tiempo, recursos y potencia de procesamiento afectan la posibilidad de conseguir los resultados esperados [3].

Los documentos manuscritos, además, suelen presentar marcas realizadas durante y luego del momento de su composición (correcciones, borrones, tachaduras, llamados, sellos, marcas, etc.) o detalles propios del paso del tiempo (desgaste, manchas), que hacen más difícil su legibilidad e interpretación, ya sea porque no nos permite interpretar correctamente el contenido del manuscrito o porque dicha marca dota al texto de un contexto (histórico, circunstancial, emocional, etc.) en el cual fue escrito, generando situaciones en las que estas marcas determinan el significado de los documentos. Otros elementos se encuentran mas relacionados con la estructura del manuscrito: tablas, listados, texto en columnas y texto en orientaciones no convencionales. Por lo que el reconocimiento de estos tipos de textos continúa siendo un desafío y, al depender de la supervisión humana para garantizar la calidad y la fiabilidad en los textos obtenidos [4], hace que se descarte la idea de utilizar técnicas de reconocimiento automático.

En este marco se podría contemplar el uso del método de transcripción “manual” de manuscritos, donde el transcriptor realiza la interpretación de lo escrito por el autor, para luego ingresarlo en algún procesador de texto. Esta tarea suele resultar tediosa y propensa a errores ya que el transcriptor puede no lograr comprender el contenido de los documentos. En este contexto es clave la participación de otros integrantes que puedan agilizar la toma de decisiones y contribuir en el proceso, dando lugar así a un enfoque colaborativo de la transcripción.

Para el óptimo desarrollo de este enfoque se considera crucial disponer un espacio propicio donde se pueda organizar el trabajo en conjunto así como también contar con herramientas aptas para llevarlas a cabo. Cuantificando esta propuesta, el uso de técnicas de colaboración utilizadas por la ciencia ciudadana, como el crowdsourcing, permitiría a una comunidad hacer participe de proyectos de esta índole, generando intercambio, optimizando y enriqueciendo el proceso de transcripción.

Como punto adicional, en este enfoque colaborativo de transcripción, se debería plantear un método para mantener el interés y alentar a que los involucrados colaboren en los proyectos. Una posibilidad es analizar distintas formas de ludificación de las tareas, para que los participantes enfrenten desafíos en juegos, cumplan objetivos y así obtener logros e insignias potenciando y haciendo más ameno el proceso presentado.

1.2. Objetivo

Desarrollar una plataforma colaborativa de transcripción con el fin de que los miembros de una comunidad puedan cargar manuscritos digitalizados y colaborar en la transcripción de los mismos, definiendo diversas capas de transcripción las cuales serán representadas utilizando tecnologías de la web semántica, para que puedan compartirse y a su vez puedan ser procesadas por autómatas.

1.3. Objetivos específicos

- Crear una plataforma en la cual su comunidad podrá expresar sus opiniones sobre las distintas transcripciones y sobre los términos que se adecuan más a lo que el texto quiere transmitir y, de esta manera, generar las transcripciones de manera comunitaria.
- Incorporar marcas de referenciación de manera que se pueda corresponder el texto transcrito con el fragmento del manuscrito digitalizado.
- Agregar un método de votación de transcripciones y en base a este método seleccionar la transcripción más adecuada para un fragmento del manuscrito.
- Creación de foros de discusión sobre los cuales los usuarios podrán discutir e intercambiar opiniones sobre cada transcripción. Integrar el concepto de gamificación sobre todo el proceso de transcripción.
- Integrar a LOD (linked open data) para que la información generada pueda ser consumida por servicios en la web semántica y así enriquecer la información sobre estas transcripciones.
- Definir capas de transcripción, ya sean cronológicas, de identidad o descriptivas, sobre un documento dado, de manera que pueda incluirse una representación digital lo más cercana posible al original digitalizado.
- Indexar los textos transcritos de manera que cualquier usuario del sitio pueda consultar y realizar búsquedas sobre la base de datos de conocimiento referenciando lugares, géneros, versiones e idiomas.

1.4. Estructura del Documento

- **Capítulo 1 - Introducción**
En este capítulo se plantea la motivación principal de la tesina, la problemática a la que hace referencia y los objetivos propuestos.
- **Capítulo 2 - Transcripción**
En este capítulo se detallará una introducción al concepto de transcripción junto con las maneras que se puede realizar y las distintas tecnologías existentes para llevarse a cabo.
- **Capítulo 3 - Marco conceptual**
En este capítulo se profundizará en la teoría de los conceptos que dan fundamentos a Transcriptor como la ciencia ciudadana y el concepto de gamificación, además se describen las tecnologías utilizadas como la web semántica y ontologías.
- **Capítulo 4 - Trabajo relacionado**
En este capítulo se detallarán algunos proyectos de transcripción actuales, como el caso de FromThePage, Zooniverse, Scribe y Recogito, que fueron de inspiración y utilizados como referencia para la tesis planteada.
- **Capítulo 5 - Estrategia general**
En este capítulo, se presentará el desafío y el diseño de la solución que enmarca esta tesis, cuales son los objetivos de negocio a los que se aspira a resolver y cuales son las características principales de la solución propuesta.
- **Capítulo 6 - Arquitectura e implementación**
En este capítulo, se presentará a nivel técnico la arquitectura de Transcriptor, el detalle de su implementación, las tecnologías utilizadas y documentación para el desarrollo. Además se detallarán los pasos para la instalación y una breve descripción de los principales flujos de la plataforma.
- **Capítulo 7 - Evaluación y presentaciones**
En este capítulo se abordará el proceso de construcción y realización de pruebas sobre la plataforma en distintos ámbitos y utilizando métodos de evaluación tanto empíricos como técnicos. También se incluyen las presentaciones en congresos y las demostraciones públicas del uso del sistema.
- **Capítulo 8 - Conclusiones y trabajo futuro**
En este capítulo se expondrán las conclusiones obtenidas a partir del trabajo realizado en esta Tesis, y se presentarán trabajos a futuro sobre el área de investigación de la misma.

Parte I

Marco conceptual

Capítulo 2

Transcripción

2.1. Transcripción

Una definición del término transcripción es útil para comprender algunos de los elementos involucrados en el proceso y permite analizar cómo y porqué transcribir documentos digitalizados.

Según la Real Academia Española, el término *Transcribir* encuentra su significado en “copiar, escribir en una parte lo escrito en otra”. Con Lo cual, a pesar de que esta expresión es correcta ya que el término puede utilizarse como un sustituto a la acción *Transliterar*, la palabra *transcripción* ha tomado diversas definiciones según el campo en el cual ha sido utilizado y adquiriendo conceptos que la llevan más allá de referirse a una simple copia textual de un contenido escrito.

Algunos ejemplos de estos múltiples significados se pueden encontrar en la genética, donde el término *transcripción* refiere al proceso de traducción del ADN en el cual este se sintetiza en ARN a través de la encima polimerasa. En música se utiliza para describir el proceso de adaptación de una pieza pensada para un instrumento en otro u otros instrumentos. En lingüística se hace uso del termino para representar la técnica de “convertir” un discurso dado, una fotografía o manuscrito, en puro texto [1].

De esta manera y analizando las diferentes contextualizaciones que se fueron planteando para el término, podemos elaborar una definición un poco más completa para este concepto: La transcripción identifica un proceso, un producto y una relación entre documentos, en el cual dicho proceso se encarga no solo de volcar el contenido literal de un documento A al B, sino también de procesar esta información para que en el contexto de B estos datos puedan ser útiles, respetando el contenido conceptual y objetivos de A.

En esta tesis nos enfocaremos en la transcripción de manuscritos y documentos, la cual históricamente ha sido utilizada como una herramienta de documentación y seguimiento de información, ayudando a registrar y transmitir conocimientos en campos complejos como lo son la medicina y las leyes, hasta procesos gubernamentales o comerciales.

Con la era de la informática, la transcripción se ha modernizado, dando lugar a la digitalización de los documentos, que consiste en la toma de imágenes de los documentos y/o la creación de archivos de texto con el contenido de los documentos físicos.

2.2. Modos de transcripción

En la confección de una transcripción de un texto dado existen múltiples factores a tener en cuenta para preservar una estructura coherente, los cuales dependerán mucho del nivel de detalle que se desea conservar y del objetivo que se desea alcanzar realizando

dicha transcripción. Según Tanodi [5] esta diversidad en el modo de transcribir documentos históricos manuscritos tiene que ver con dos tendencias, las cuales los copistas y editores adoptaron a través del tiempo, que caben mencionar y analizar a continuación:

- **Transcripción Literal:** Busca transcribir de forma fiel al manuscrito en su aspecto paleográfico, Tanodi sostiene que la misma si bien “fue la más usada en las ediciones de las últimas décadas por muchos historiadores, archivos, institutos históricos y bibliotecas [...] no es la más acertada porque ha causado, por ejemplo, muchos problemas y distorsiones en la interpretación de las abreviaturas”.
- **Transcripción Modernizada:** Este modo de transcripción intenta mantener un enfoque más práctico para la comprensión del lector, donde la ortografía y terminología original de los documentos es sustituida por la contemporánea, con lo cual la fonética de los documentos se ve afectada para proveer mayor accesibilidad al contenido.

Sobre estas corrientes, surge un tercer enfoque intermedio, el cual intenta mantener las virtudes de los modos presentados:

- **Transcripción Literal Modernizada:** Esta tendencia busca mantener las características de los enfoques literal y modernizado, esforzándose en presentar el contenido del manuscrito de manera accesible para el lector respetando la fidelidad al texto del manuscrito, la ortografía y terminología antigua se mantienen, pero a su vez se sustenta la transcripción utilizando ediciones impresas, que permiten una mayor facilidad de lectura y accesibilidad al lector, para de esta forma analizar tanto de manera paleográfica como literal el contenido del documento.

2.3. Tecnologías de transcripción de textos

Cuando se dispone de información en forma de documento impreso y se desea procesar mediante una computadora, existen dos opciones.

La primera consiste en la introducir el contenido del texto manualmente a través del teclado, la que suele ser una labor larga y tediosa, pero con resultados de gran precisión. Por otro lado, la segunda alternativa busca automatizar esta operación por medio de un sistema de reconocimiento de caracteres, compuesto de un software y hardware adecuado que reduciría considerablemente el tiempo de entrada de datos, con ciertos costes en la precisión de los datos obtenidos.

En la actualidad es posible automatizar la conversión de estos datos con varias técnicas que permiten transcribir textos digitalizados partiendo de imágenes estáticas a texto editable. Algunas de ellas son:

- **OCR (Optical Character Recognition):** Esta tecnología permite reconocer automáticamente caracteres a través de un mecanismo óptico, OCR puede reconocer tanto texto escrito a mano como impreso. Pero su rendimiento depende directamente de la calidad de documentos de entrada y es por esto que se sugiere para procesar imágenes que consisten casi en su totalidad en texto y con correcta alineación en cuanto a los caracteres que conforman las palabras y líneas del documento [6]. Cualquier libro u hoja impresa en una imprenta se puede escanear como imagen y convertida a texto gracias a esta tecnología.
- **ICR (Intelligent Character Recognition):** usualmente utilizado para formularios de carga manuscrita, la tecnología IRC se vale de elementos funcionales impresos en el

1. Preference for station applied for . (Refer Table 1)

Ist B A N G A L O R E
Ist C H E N N A I
Ist H Y D E R A B A D

2. Item code (Refer Table 2):

E L E C T R O N I C

3. Name(Shri/Smt/Kum):

Figura 2.1: Ejemplo de formulario con marcas para el reconocimiento ICR

papel para procesar las entradas de texto escrito, tales como pueden ser recuadros para completar, círculos para rellenar o recuadros para tachar. Luego estos documentos se digitalizan (utilizando un scanner) y se procesan utilizando redes neuronales, las cuales comparan los trazos dispuestos sobre el contenedor para poder determinar cual es el caracter escrito. Esta técnica no es aplicable a escritura cursiva, ya que depende de los espacios entre letra y letra para analizar la información [7].

- HTR (Handwritten Text Recognition): esta tecnología lleva el reconocimiento de texto a un nuevo nivel de complejidad, definiendo nuevos procesos para realizar el trabajo. Es así como dentro de este tipo de tecnologías pueden distinguirse dos enfoques:
 1. Enfoque automático: esta técnica utiliza un preprocesamiento del documento digitalizado, el cual detecta las líneas del manuscrito para definir y alinearse correctamente sobre las palabras escritas. Luego, un segundo procesamiento separa las líneas en fragmentos de una palabra para finalmente, en un ultimo proceso, reconocer cada palabra de las secciones individuales extraídas.
 2. Enfoque asistido: Este enfoque puede representar tanto una alternativa como un complemento al Enfoque Automático. De esta manera, a través de hardware como un lápiz, tableta digitalizadora o panel táctil el transcriptor deberá reescribir las palabras siguiendo sus trazos sobre el manuscrito digitalizado, brindando información sobre cómo cada palabra esta “dibujada” sobre el papel y aportando conocimiento para deducir los caracteres del texto.

Con esta tecnología es posible generar una transcripción con un nivel importante de automatización, pero con una tasa de errores que van del 20 % al 60 %, con lo que es necesaria de un proceso de una persona que revise los resultados para cerciorarse que sean los esperados [8].

Sumado a los inconvenientes ya mencionados en el reconocimiento automatizado de documentos manuscritos, cabe mencionar que este tipo de documentos suelen presentar marcas realizadas durante y luego del momento de su composición (correcciones, borrones, tachaduras, llamados, sellos, marcas, etc.) o detalles propios del paso del tiempo (desgaste, manchas), que hacen más difícil su legibilidad e interpretación, ya sea porque dificulta la correcta interpretación del contenido del manuscrito o porque dicha marca dota al texto

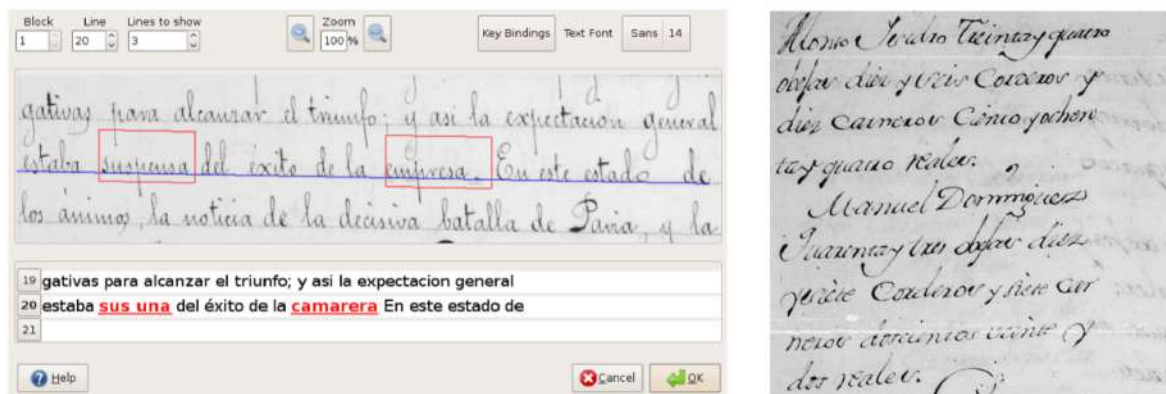


Figura 2.2: Ejemplo de procesamiento de textos en HTR

de un contexto (histórico, circunstancial, emocional, etc.) en el cual fue escrito, generando situaciones en las que estas marcas determinan el significado de los documentos. Otros elementos se encuentran más relacionados a la estructura del manuscrito: tablas, listados, texto en columnas y texto en orientaciones no convencionales. Por lo que reconocimiento de este tipos de textos continúa siendo un desafío y que sigue dependiendo de la supervisión humana para garantizar la calidad y la fiabilidad en los textos obtenidos. [4]

2.4. Transcripción aplicada a la ciencia ciudadana

Teniendo en cuenta los contenidos tratados hasta ahora se puede afirmar que la transcripción de manuscritos ayuda a acortar la brecha entre la digitalización de textos y su estudio al mejorar el acceso a través de la búsqueda de texto completo, enriquecer la colección de metadatos y abrir las colecciones al análisis textual digital.

Los proyectos de transcripción requieren de mucho tiempo y son intelectualmente intensivos y costosos para una organización. Esto motivó el surgimiento de modelos, entornos y plataformas de trabajo colaborativo (crowdsourcing), donde los beneficios de la transcripción digital empezaron a cobrar importancia y desarrollar un papel fundamental en este campo [9].

Los modelos de colaboración en transcripción persiguen la meta de permitir que una comunidad aporte, mantenga y se retroalimente de contenido, facilitando su utilización para diversos fines como mediciones, análisis, investigaciones y estudios.

El hecho de poner la colaboración como principal vía de generación e intercambio de datos llevó a las instituciones y organizaciones a abrir la convocatoria de participación a personas no profesionales y aficionados al tema para aumentar la masa de colaboradores, mejorando la recolección de datos y ampliando la gama de resultados obtenidos incluyendo no sólo las ideas de un equipo designado para la labor, sino la combinación de conocimientos que un grupo numeroso de personas puede aportar.

Este enfoque colaborativo no es novedoso, sino que forma parte del concepto de ciencia ciudadana. Esta se define como investigaciones científicas llevadas a cabo por una suma de colaboradores, ya sean científicos y profesionales o personas comunes. Donde el público en general contribuye a la recopilación, análisis y comprensión de datos científicos [10].

Aplicado a este contexto, el crowdsourcing emerge como uno de los pilares fundamentales para el desarrollo de la ciencia ciudadana, proponiendo un modelo sostenible para

la generación de transcripciones. Esta colaboración abierta y diversa no solo impulsa la investigación científica, sino que también fomenta la participación activa de la comunidad, democratizando el proceso científico y enriqueciendo la labor de expertos con la contribución valiosa de individuos no especializados.

Capítulo 3

Web Semántica

3.1. Definición

La web semántica [11] es un proyecto que fue propuesto por Tim Berners-Lee (el inventor de la world wide web) con el propósito de añadir a la web actual metadatos semánticos para dar un significado explícito a los recursos de la web para que las máquinas puedan procesar y comprender su contenido, ahorrando trabajo a las personas en la búsqueda y optimizando los resultados.

En la Web Semántica, la información no solo está destinada a ser leída por humanos, sino también interpretada por máquinas. Esto se logra mediante el uso de estándares y tecnologías que permiten asignar significado preciso a los datos. Algunos de los elementos clave de la Web Semántica incluyen: Ontologías, RDF y OWL.

3.2. Componentes de la web semántica.

3.2.1. Ontologías

En el contexto de la web semántica, una ontología [12] se define como “una especificación explícita y formal sobre una conceptualización compartida”. Una interpretación de esta definición es que una ontología es una especificación explícita de un concepto en un dominio y las relaciones entre diferentes conceptos. Estas ontologías proporcionan un marco formal para describir el conocimiento y la semántica de un dominio particular. Una ontología proporciona un vocabulario de clases y relaciones para describir un dominio, poniendo el acento en la compartición del conocimiento y el consenso en la representación de éste. Por ejemplo, una ontología sobre arte podría incluir clases como Pintor, Cuadro, Estilo o Museo, y relaciones como autor de un cuadro, pintores pertenecientes a un estilo artístico u obras localizadas en un museo. Algunos ejemplos de ontologías son FOAF, DBPedia y Schema.org.

3.2.2. RDF y RDFS

RDF [13] (Resource Description Framework) es un estándar de la W3C diseñado para facilitar la representación de recursos y metadatos en la web. El elemento de construcción básica en RDF es el “tripleta” o sentencia, que consiste en dos nodos (sujeto y objeto) unidos por un arco (predicado), donde el nodo sujeto representa el recurso que se describe, el arco (o predicado) representa la relación o propiedad del recurso y el objeto representa el valor o destino de la propiedad, el cual también puede ser otro recurso.

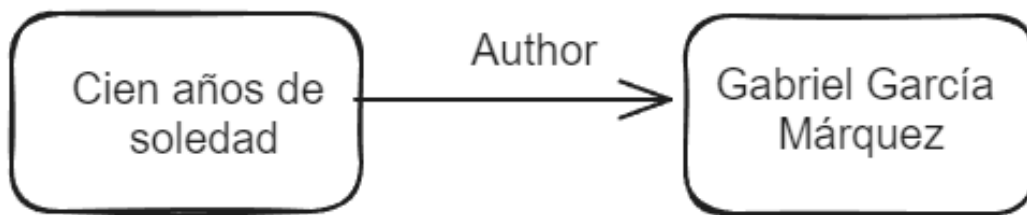


Figura 3.1: Ejemplo RDF

Los componentes clave de RDF son los recursos, que necesitan poder representarse de forma única en la web, de esta manera las URIs representan un papel fundamental en RDF para la representación tanto de recursos como las propiedades. Por ejemplo una sentencia podría expresar el hecho de que el autor (predicado) del libro “Cien años de soledad” (sujeto) fue el escritor Gabriel Garcia Marquez (objeto).

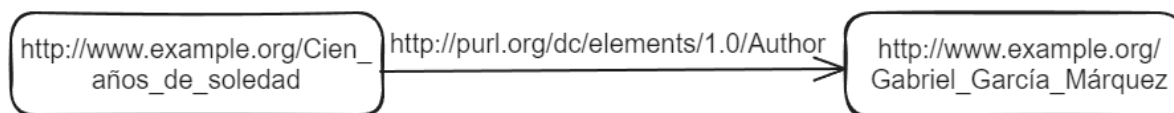


Figura 3.2: Ejemplo RDF

RDF provee una manera simple de representar recursos y propiedades, sin embargo, por si solo no se puede indicar si un recurso pertenece a una clase o si es de algún tipo. RDF Schema (RDFS) es una extensión de RDF, donde se pueden definir jerarquías de clases de recursos, especificando las propiedades y relaciones que se admiten entre ellas. En RDF las clases, relaciones, y las propias sentencias son también recursos, y por lo tanto se pueden examinar y recorrer como parte del grafo, o incluso acertar sentencias sobre ellas.

3.2.3. OWL

OWL [12] (Web Ontology Language) es un lenguaje de ontologías diseñado para la web semántica. Las ontologías son estructuras que definen conceptualmente los términos y las relaciones entre ellos en un dominio específico. OWL, desarrollado por el World Wide Web Consortium (W3C), extiende y mejora las capacidades de RDF y RDF Schema para proporcionar un marco más rico y expresivo para la descripción de ontologías.

3.2.4. Linked Data

Linked data (LD) [14] es un concepto que hace referencia a aquellos datos publicados de acuerdo con los principios diseñados para facilitar los vínculos entre conjunto de datos, vocabularios y otros elementos. Linked Data utiliza URIs (uniform resource identifiers) como identificadores únicos globales para todo tipo de recursos. Los datos enlazados se expresan utilizando estándares como RDF.

3.2.5. JSON-LD

JSON-LD [15] es una sintaxis ligera para serializar Datos enlazados (LD) derivada de JSON que permite que un objeto JSON pueda interpretarse como un objeto Linked Data , tomando las ventajas que tiene JSON como la legibilidad y su facilidad de uso. Además de todas las características que JSON introduce, JSON-LD agrega las siguientes:

- Un mecanismo de identificación universal para objetos JSON mediante el uso de IRI (Internationalized Resource Identifier).
- Una forma de eliminar la ambigüedad de las claves compartidas entre diferentes documentos JSON asignándolas a IRI a través de un contexto
- Un mecanismo en el que un valor en un objeto JSON puede hacer referencia a un objeto JSON en un sitio diferente en la Web.
- La capacidad de anotar cadenas con su lenguajes.
- Una forma de asociar tipos de datos con valores como fechas y horas.

JSON-LD y RDF

JSON-LD es una sintaxis RDF concreta, por lo tanto un documento JSON-LD es tanto un documento RDF como un documento JSON y representa una instancia de un modelo de datos RDF. Sin embargo, JSON-LD también amplía el modelo de datos RDF para permitir opcionalmente que JSON-LD serialice conjuntos de datos RDF generalizados, estas extensiones son:

- En JSON-LD las propiedades pueden ser IRIs o nodos en blancos mientras que en las propiedades RDF (predicados) tienen que ser IRIs.
- En JSON-LD las listas son partes del modelo de datos mientras que en RDF son parte del vocabulario.
- Los valores de RDF son literales escritos o cadenas etiquetadas por lenguajes mientras que JSON-LD también soporta tipos de datos nativos de JSON como números, cadenas de caracteres y booleanos.

En resumen, estas diferencias muestran que JSON-LD es capaz de serializar cualquier conjunto de datos de RDF pero no todos.

Capítulo 4

Gamificación

4.1. Definición y beneficios

La ludificación, también conocida como gamificación, engloba la aplicación de tácticas, componentes y dinámicas inherentes a los juegos y el entretenimiento en contextos no necesariamente recreativos. Su propósito radica en potenciar la motivación y fortalecer conductas orientadas a la resolución de problemas, la mejora de la productividad, el logro de objetivos, la facilitación del aprendizaje y la evaluación de individuos específicos. Al incorporar elementos lúdicos en entornos no asociados tradicionalmente con el juego, la ludificación busca impulsar un compromiso más profundo y efectivo en diversas áreas de la vida y el trabajo [16] [17].

La gamificación aprovecha los recursos de los juegos, tales como la atracción y la afición para mantener el interés del participante en las tareas que demanda la aplicación. Esto no significa que la gamificación convierta toda la dinámica del sistema en un juego, sino más bien colabora en sus flujos de negocio motivando al usuario durante el proceso [18]. La atención del participante obtenida es canalizada para comprometerlo con las actividades, profundizar conocimientos, mejorar sus habilidades o bien, se persigue un bien comercial (customer engagement) [19].

Un ejemplo de metodología de gamificación es brindar recompensas por realizar acciones dentro del sistema. Estas pueden ser puntos, logros o insignias. Acciones como “Ingresar por primera vez”, “Compartir con otros usuarios” o “Colaborar N cantidad de veces” podrían ser iniciadores para la adjudicación de una recompensa.

Teniendo en cuenta esta definición y ejemplos, se puede enumerar algunas de las motivaciones de la implementación de la gamificación en una aplicación [19]:

- Incentivar la motivación del aprendizaje, ya sea pragmático en el uso del sistema, o mas bien teórico sobre los contenidos que se desean transmitir.
- Brinda al usuario una retroalimentación constante sobre su progreso, estableciendo una ruta de aprendizaje haciéndole saber cual es su experiencia dentro de la plataforma.
- Logra un aprendizaje mas significativo, facilitando la retención de información con el atractivo de alcanzar nuevos hitos de experiencia en la plataforma.
- Del mismo modo, con el afán de lograr nuevos hitos, el usuario mantiene un mayor compromiso con las tareas y contenido tratado.

- El aprendizaje y uso en la plataforma generan métricas cuantificables. Pudiendo analizar la experiencia de una comunidad según el puntaje, insignias o logros que posean en la plataforma.
- Genera un nivel de competencia adecuada entre colaboradores, mejorando progresivamente y en conjunto sus aportes para lograr los objetivos planteados.
- La competencia establecida por los rangos de experiencia estimula a que los participantes se vuelvan mas autónomos y autodidactas, buscando nuevas formas de aprender y conseguir retribuciones que aceleren su crecimiento en la plataforma.
- Esta competencia, a su vez, genera una mayor colaboración en la plataforma.
- Potencia la interacción entre usuarios de la comunidad.

Capítulo 5

Trabajo Relacionado

La transcripciones de manuscritos digitalizados no son un tema nuevo en el campo informático. En este capítulo se presentan proyectos que han servido de inspiración para la concepción del presente trabajo y otros casos de éxito con enfoques orientados a la colaboración.

5.1. FromThePage

FromThePage ¹ es un sitio web de transcripción y traducción colaborativo de documentos digitalizados de código libre donde los miembros de la comunidad pueden cargar manuscritos en archivos de imágenes o PDF y colaborar en la transcripción o traducción de los mismos, brindando al usuario un entorno simple para realizar este trabajo.

Organización de los proyectos

FromThePage organiza el contenido cargado de manera jerárquica en distintas secciones que actúan de contenedores.

- **Colecciones:** Una colección es un conjunto de documentos (Trabajos) que se han reunido para formar una unidad. Cada obra debe pertenecer a una colección, cada página a una obra. Las colecciones y obras se pueden traducir y transcribir.
- **Trabajo o Proyecto:** Un trabajo es un elemento individual (libro, artículo, diario, carta, etc.) que se carga en una colección. Un trabajo solo puede pertenecer a una sola colección.
- **Páginas:** Una página es la unidad más pequeña dentro de una obra (como páginas de un libro, artículo, revista, etc.) Es una sola imagen de un trabajo, generalmente una sola página dentro de un libro o una carta, aunque ocasionalmente un escaneo puede abarcar Múltiples páginas en una imagen.

Usuarios y Roles

Los usuarios en FromThePage se dividen en 3 roles:

- **Administrador:** Es el usuario que se encarga de la administración del sitio, puede gestionar colecciones, proyectos y ver estadísticas de uso del sistema.

¹<http://fromthepage.com/>

- **Propietario:** Son aquellos usuarios con acceso privilegiado sobre los proyectos y colecciones debido a que fueron quienes cargaron el trabajo inicialmente. Además de gestionar el proyecto, controlan los permisos de acceso de los usuarios sobre el mismo. Por defecto, una colección es visible para cualquier persona que acceda a FromThePage desde internet. Una colección puede tener más de un propietario.
- **Colaboradores:** Son los usuarios que se encargan de ayudar en las transcripciones o traducciones de una colección o un trabajo.

Los usuarios pueden cargar sus documentos de dos formas. La primera se realiza creando una colección individual y luego agregando sus correspondientes trabajos y páginas. El segundo método es mediante la subida de archivos zip, los cuales pueden contener carpetas que agrupan imágenes, archivos PDF o incluso otras carpetas con archivos. FromThePage respeta el agrupamiento realizado en el archivo comprimido y lo replica utilizando las colecciones y proyectos según los nombres de los directorios recibidos. Una vez creada la colección, el propietario de esta puede modificar la visibilidad a colaboradores y otros propietarios para generar restricciones o autorizando el acceso a las transcripciones de los trabajos, lo cual facilita a gestión de equipos durante el proceso de transcripción.

Sobre la transcripción

Como se mencionó con anterioridad, cualquier usuario registrado en FromThePage podrá transcribir documentos, la transcripción de los documentos tienen las siguientes características:

- **Edición según el estilo wiki:** Los usuarios agregan o editan transcripciones usando una sintaxis simple de estilo wiki en un lado de la pantalla mientras ven una imagen escaneada de la página del manuscrito en el otro lado.
- **Control de versiones:** Los cambios en la transcripción de cada página se registran y se pueden ver para seguir el historial de edición de una página.
- **Wikilinks:** Los temas mencionados en el documento pueden indexarse mediante simples wikilinks dentro de la transcripción. Los usuarios pueden anotar temas con artículos de temas completos.
- **Presentación:** Los lectores pueden ver las transcripciones en un formato de varias páginas o junto a las imágenes de la página. También pueden leer todas las páginas que mencionan un tema.
- **Marcado automático:** FromThePage puede sugerir wikilinks a editores mediante la extracción de transcripciones editadas previamente. Esto ayuda a asegurar la coherencia editorial y reduce enormemente la cantidad de esfuerzo involucrado en el marcado.
- **Integración de Internet Archive:** FromThePage puede apuntar a manuscritos alojados en Archive.org. Importará la estructura de la página y los títulos de las páginas impresas en su formato nativo para la transcripción, al tiempo que publicará imágenes de la página del Archivo de Internet.

Una vez que terminada la transcripción, el propietario marcará la página como “transcripta” finalizando así la transcripción de la misma.

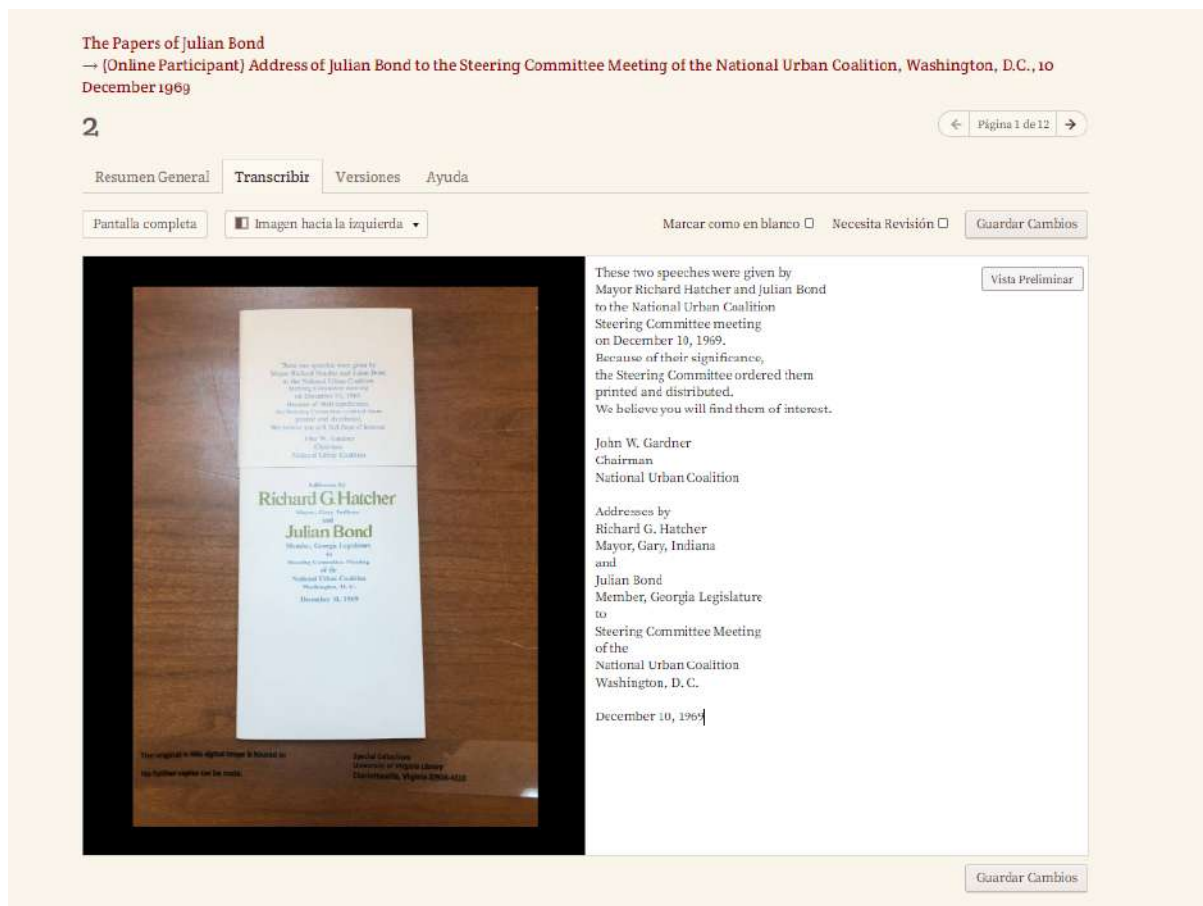


Figura 5.1: Ejemplo del panel de transcripción de FromThePage

5.2. Zooniverse

Zooniverse² es una plataforma web de ciencia ciudadana que actualmente cuenta con miles de voluntarios registrados que trabajan en colaboración con investigadores profesionales en múltiples proyectos de investigación en una amplia variedad de disciplinas, desde astronomía hasta biología, climatología y temas de humanidades.

Los voluntarios podrán estudiar objetos auténticos de interés reunidos por investigadores tales como imágenes de galaxias, registros y diarios históricos o vídeos de animales en sus hábitats naturales. Al responder preguntas simples sobre estos objetos, ayudará a contribuir a nuestra comprensión de nuestro mundo, nuestra historia, nuestro universo.

Si bien Zooniverse puede abarcar varios temas de investigación, como lo son la exploración espacial, zoología, entre otros, se hará énfasis en su utilidad en el tratamiento de documentos históricos y transcripciones. Cualquier usuario registrado puede colaborar con la transcripción de documentos históricos, para poder resolver esto, investigadores pueden crear proyectos, definiendo un flujo de trabajo para que los usuarios interesados puedan ayudar a transcribir.

Modelo de dominio de Zooniverse

Como define en [20] el modelo de dominio es relativamente simple, y se centra en partes

²<http://zooniverse.org/>

más críticas para las operaciones de Zooniverse y la creación de instancias de proyectos, en lugar de las facetas de dominio detalladas únicas para cada proyecto:

- **User:** Las personas son fundamentales para Zooniverse. Un usuario es una entidad que puede participar en múltiples proyectos. Representa una persona / cuenta, con un nombre de usuario, una dirección de correo electrónico, e información sobre los proyectos en los que han participado.
- **Subject:** Son los elementos que los usuarios clasifican, anotan o transcriben, como por ejemplo, una imagen escaneada de un diario. Los *subjects* son lo que se muestra a los usuarios en el sitio para clasificación y están asociados con metadatos, útiles para los investigadores de un proyecto, como ubicación geográfica de origen, detalles de la cámara / telescopio, tamaño de campo, etc.
- **Workflow/Task:** Cuando a un *user* se le presenta un *subject* en uno de los proyectos, se pide que realice una Tarea. Estas tareas se pueden agrupar en un flujo de trabajo. La mayoría de los proyectos solo tienen un único flujo de trabajo. En “Notas de la naturaleza”, cada paso de la transcripción es una tarea separada, en “Galaxy Zoo”, cada paso del árbol de decisiones es una tarea también.
- **Classification:** Las clasificaciones son la unidad central de Zooniverse del esfuerzo humano. Una clasificación representa lo que persona vio y lo que dijo al respecto.
- **Group:** Los *subject* se pueden agrupar para algún propósito mas grande. Un grupo puede representar, por ejemplo una colección de manuscritos.
- **Project:** Este es simplemente el proyecto de ciencia ciudadana en sí mismo, con la cual los *subjects*, *group* y *clasification* se van a asociar.

5.2.1. Proyectos de zooniverse relacionados al mundo de la transcripción

Como se definió con anterioridad, Zooniverse es una plataforma web que permite crear proyectos de múltiples disciplinas, a continuación se detallará un proyecto que fue creado con el propósito de transcribir documentos.

Douglass Day ³ es un programa anual que conmemora y celebra el cumpleaños de Frederick Douglass, un destacado líder abolicionista, orador y escritor afroamericano del siglo XIX. Douglass nació el 14 de febrero de 1818. Este día se ha convertido en una ocasión para reflexionar sobre su legado, su contribución a la lucha por la abolición de la esclavitud y los derechos civiles, así como para destacar la importancia de la historia afroamericana. Cada año, el programa reúne a miles de personas para ayudar a crear recursos nuevos y de libre acceso para aprender sobre la historia de los negros. Con frecuencia el programa se centra en importantes archivos de mujeres negras, como Anna Julia Cooper (2020), Mary Church Terrell (2021).

Este programa organiza maratones de transcripciones, el cual se apoya en la plataforma Zooniverse, donde cada año se genera un proyecto nuevo para realizar las transcripciones de archivos. a continuación se detalla un proyecto de transcripción hecho en Zooniverse relacionado a este programa.

³<https://douglasday.org/>

5.2.2. Transcribe Cooper

Anna Julia Cooper fue una escritora, educadora y conferenciante estadounidense. Nacida en esclavitud en 1858, Se convirtió en la primer mujer afroamericana en tener un doctorado en la universidad Sorbona de Paris y se convirtió en la cuarta mujer afroamericana en obtener un doctorado. Cooper fue una figura importante en la lucha por la igualdad de derechos para las mujeres afroamericanas.

Transcribe Cooper⁴ es un proyecto de Zooniverse, con el propósito que la comunidad ayude a digitalizar y exponer los documentos y manuscritos de Cooper.

Al ingresar a la clasificación del proyecto Transcribe Cooper, se presenta un marco de trabajo como se muestra en la figura 5.2 donde se puede apreciar secciones, en el cuadro 1 se presenta una imagen que representa un archivo a clasificar (transcribir) y en el cuadro 2 tenemos 2 tabs, uno donde nos indica los pasos a realizar (el workflow definido) y el segundo indica un tutorial de como realizar el workflow.

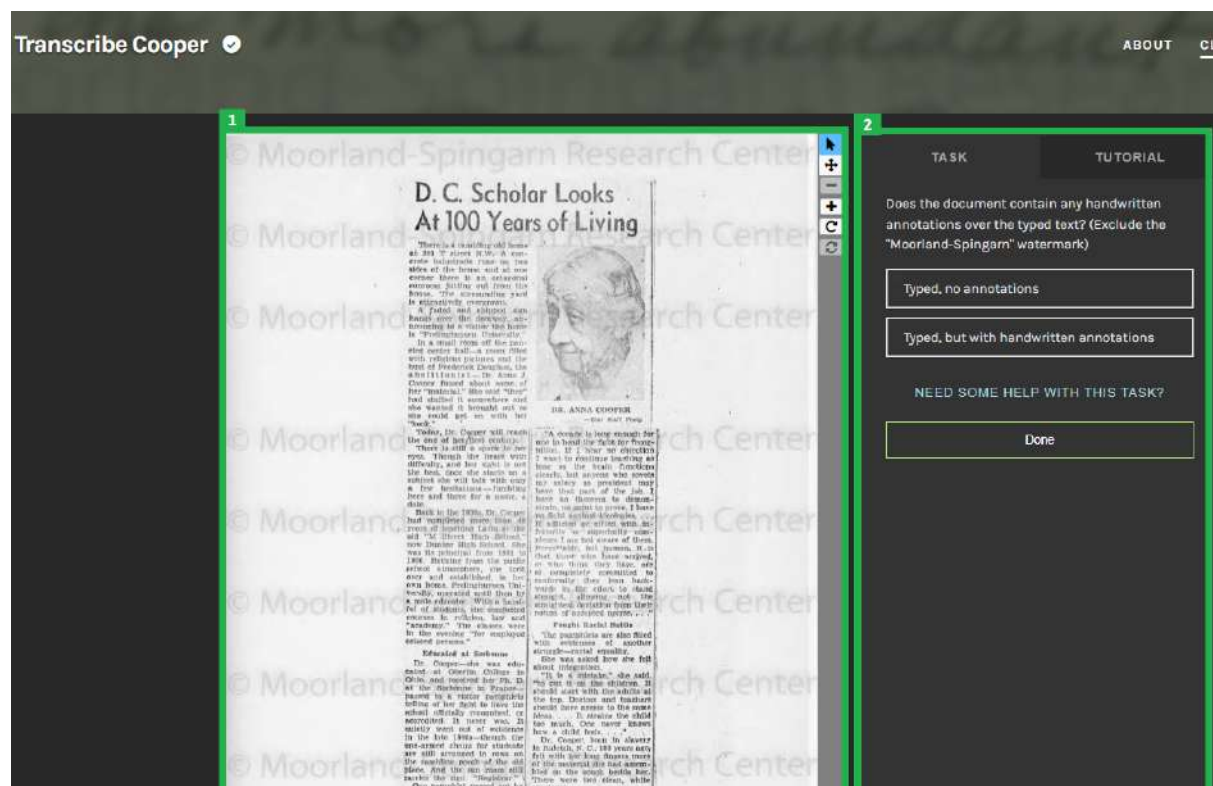


Figura 5.2: Panel de clasificación de un proyecto Zooniverse

5.2.3. Workflow de Transcribe Cooper

Transcribe Cooper tiene un workflow definido que consiste en una serie de pasos que se detalla a continuación. El primer paso consiste en seleccionar si en el documento se observa alguna nota manuscrita o no.

⁴<https://www.zooniverse.org/projects/transcribecooper/transcribe-cooper>

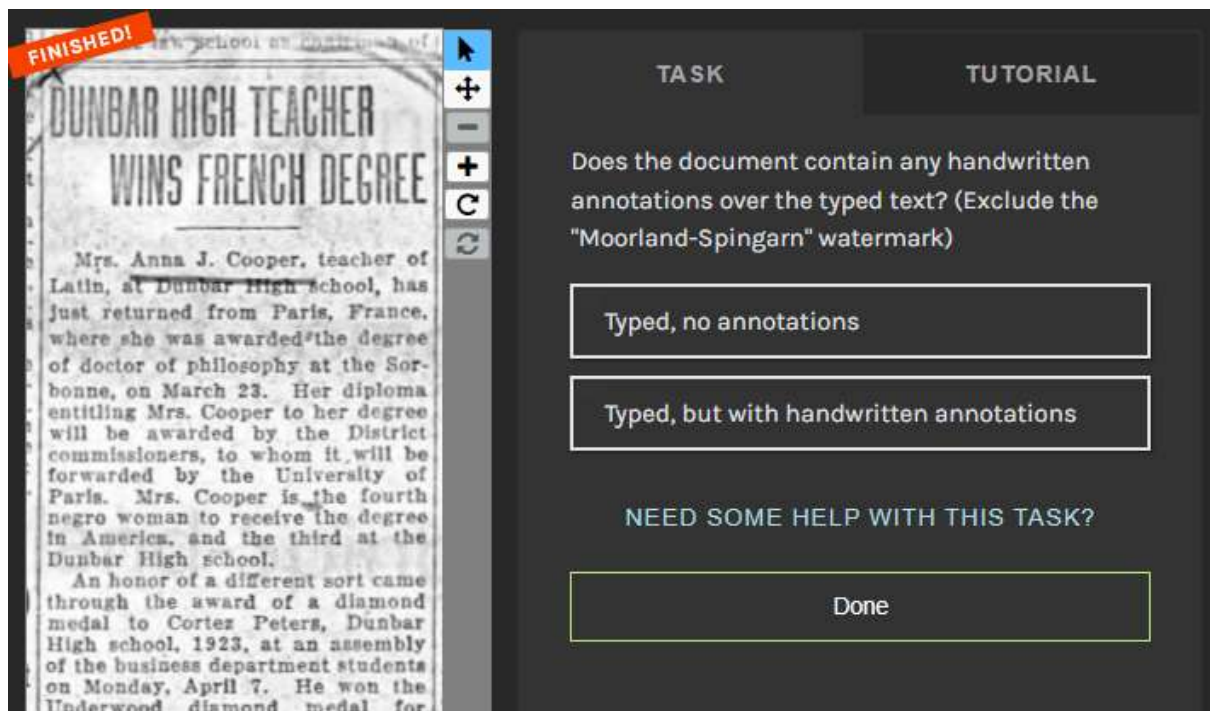


Figura 5.3: Workflow paso 1 - Selección de opciones

Si el documento no posee ninguna nota manuscrita se solicita que se transcriba lo mejor que se pueda el texto de la imagen, una vez hecha la transcripción se puede continuar presionando el botón “Done” y antes de finalizar se muestra una vista previa como se puede ver en la figura 5.5.

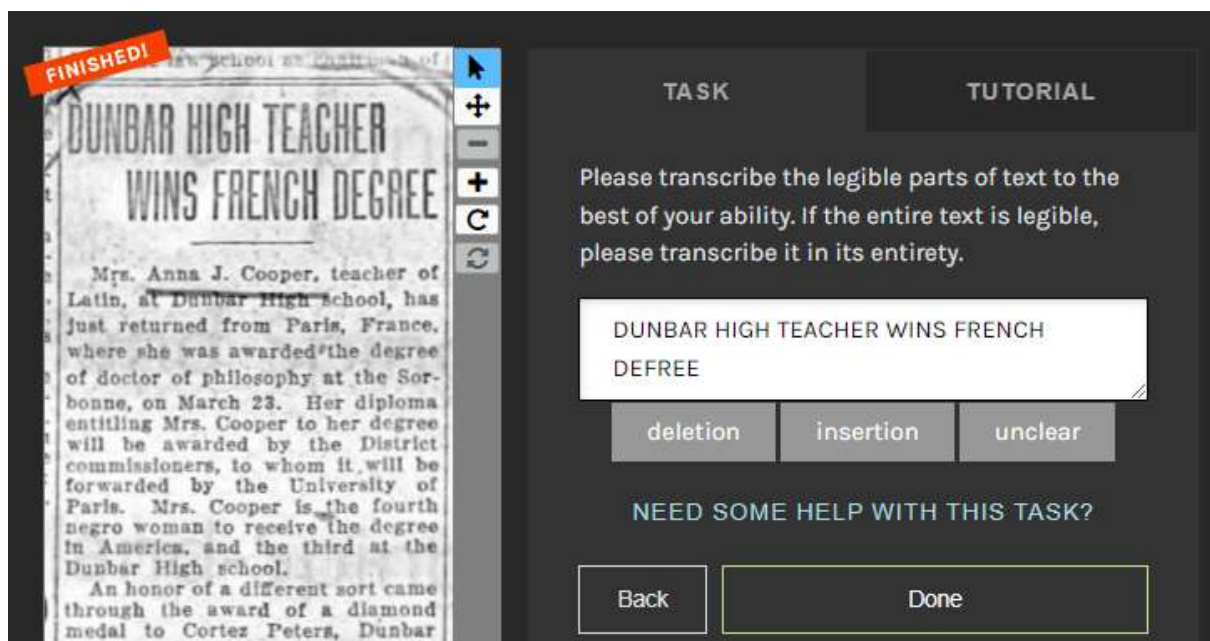


Figura 5.4: Workflow paso 2 - Cuadro de transcripción.

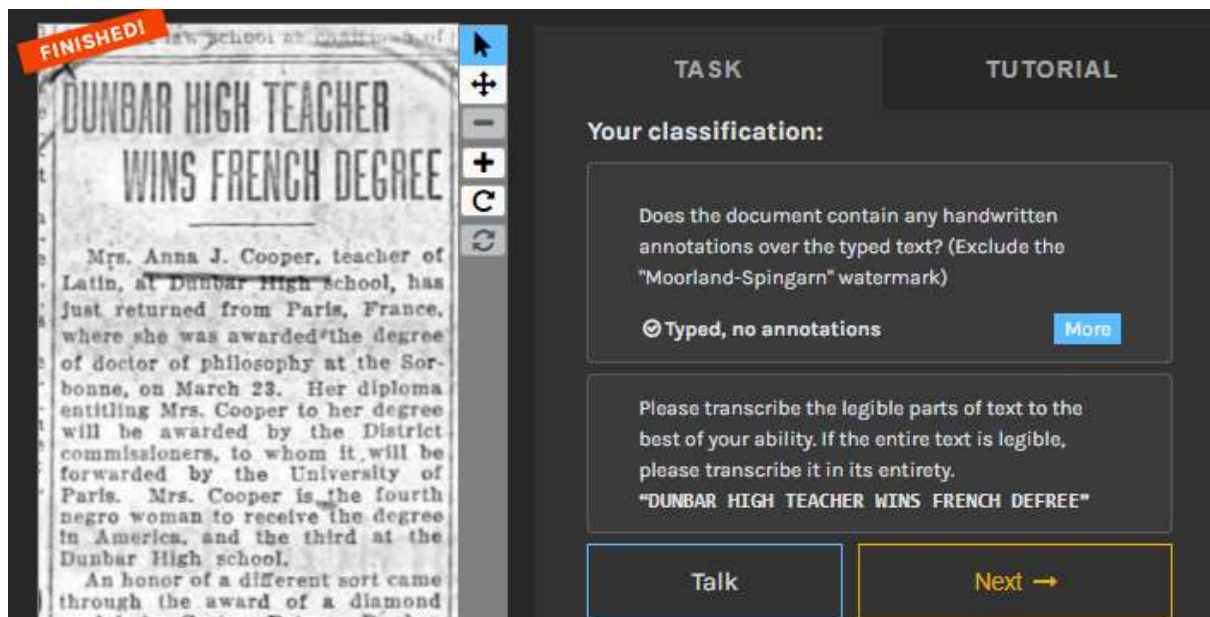


Figura 5.5: Workflow paso 3 - Vista previa.

Si el documento posee notas manuscritas, se debe seleccionar nuevamente entre dos opciones, si las notas manuscritas son legibles o no, en caso de ser legibles se pide transcribir las anotaciones, centrándose en estas e ignorando el texto escrito por maquina. y el siguiente paso sera transcribir el texto escrito por maquina, separando ambas transcripciones.

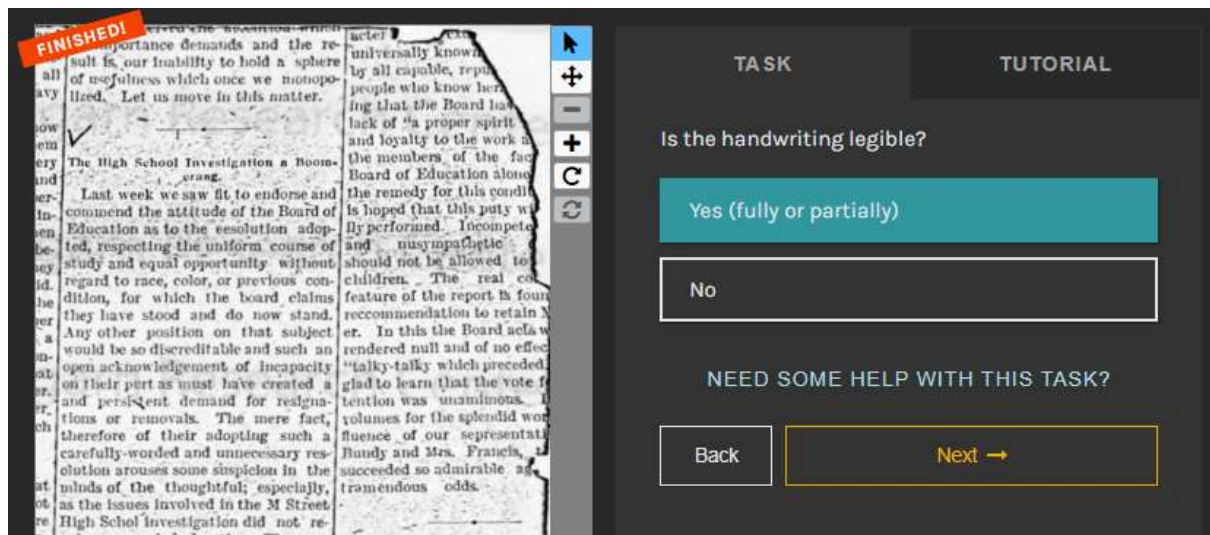


Figura 5.6: Workflow paso 4 - Selección opciones

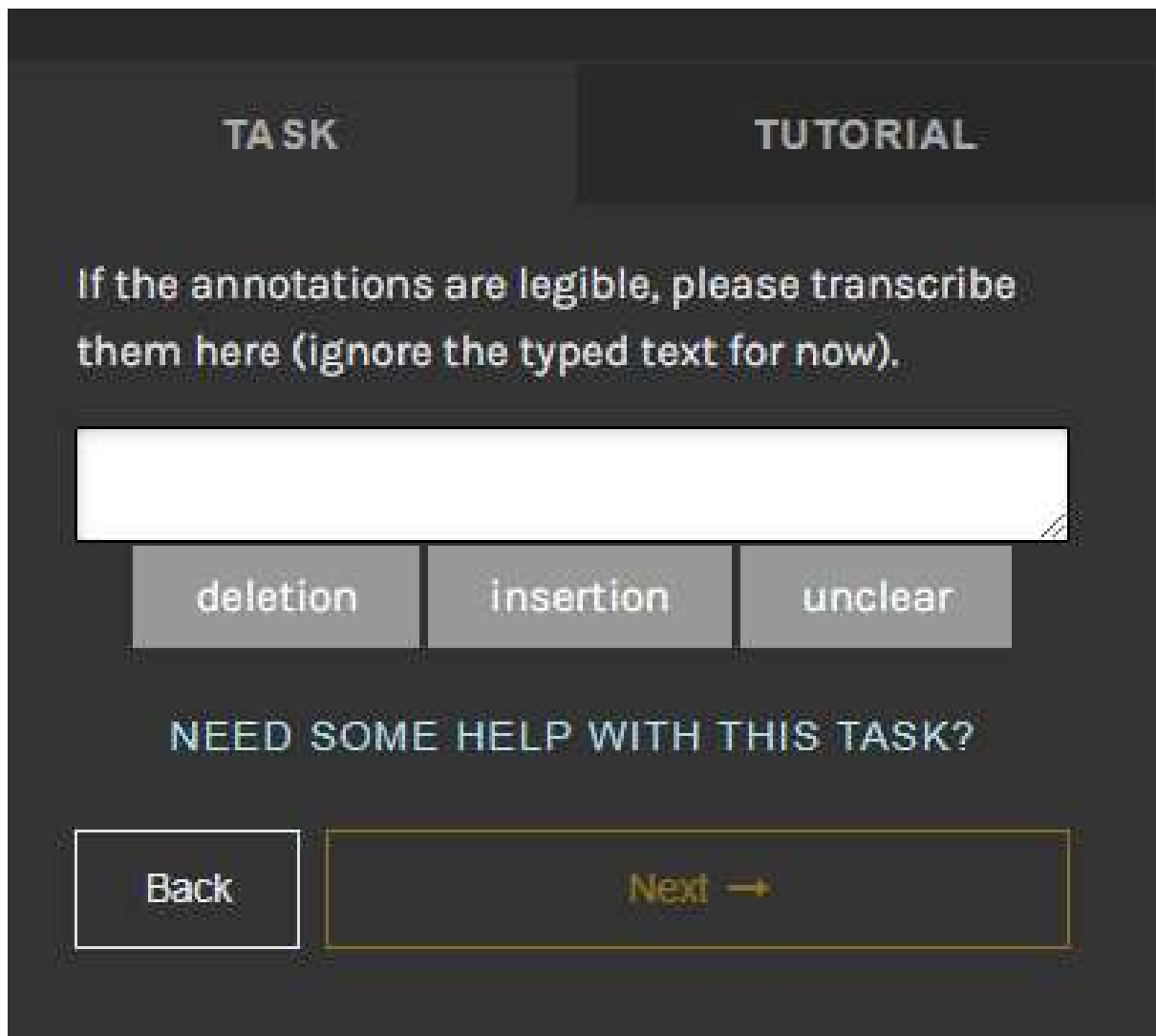


Figura 5.7: Workflow paso 5 - Cuadro de transcripción

5.3. El Proyecto Scribe

El Proyecto Scribe es un framework de código abierto creado por Zooniverse⁵ en colaboración con la Biblioteca Pública de Nueva York⁶ que permite configurar y lanzar proyectos de transcripciones de manera fácil para la comunidad (crowdsourcing), especialmente enfocados en obtener datos estructurados de materiales manuscritos o aquellos que no son accesibles mediante OCR.

El código de Scribe está basado en proyectos y prototipos anteriores producidos por Zooniverse y la Biblioteca Pública de Nueva York. Es importante destacar que Zooniverse cuenta con mucha experiencia en proyectos de crowdsourcing. El código fuente está disponible de manera pública en su repositorio de Github⁷. Además del código fuente, hay una amplia documentación que proporciona instrucciones paso a paso para los desarrolladores que buscan implementar proyectos basados en Scribe como una forma de procesar

⁵<http://zooniverse.org/>

⁶<https://www.nypl.org/>

⁷<https://github.com/zooniverse/scribeAPI>

y extraer datos de materiales fuente digitalizados.

El código y la documentación también se pueden encontrar a través de la página de inicio de Scribe⁸, junto con enlaces a los tres proyectos que NYPL y Zooniverse desarrollaron con Scribe durante el período de la subvención. Estos proyectos son: Zooniverse's Old Weather: Whaling⁹, Measuring the Anzacs¹⁰, y NYPL's Emigrant City¹¹ y muestran los flujos de trabajo altamente configurables y las interfaces de usuario intuitivas de Scribe.

5.3.1. Características de los proyectos Scribe

Los proyectos Scribe cuentan con una características similares a los proyectos Zooniverse que se describieron con anterioridad. A continuación se detallará brevemente los conceptos y jerga que se utiliza los proyectos Scribe.

- **Subjects:** Un Subject es una imagen única en la cual se les solicita a los usuarios realizar una tarea sobre ella. Algunos ejemplos de subject son:

- Una página de un diario.
- Una sola foto de un scrapbook.

Estos ejemplos se denominan Subjects primarios, denominados así porque son el punto de partida donde los usuarios comienzan a trabajar. También existe el concepto de child subjects (secondary subjects, tertiary subjects, etc), que se generan a partir de los subjects primarios.

Algunos ejemplos de los childs subjects son:

- Una pagina es una primary subject.
- Una marca ubicada en un diario que indica la ubicación de una fecha, la marca es un secondary subject.
- Un usuario transcribe la fecha que fue marcada. La transcripción es un tertiary subject.

En teoría los child subjects pueden generar child subjets adicionales de manera infinita, pero en la practica el workflow de Scribe se genera sobre tres descendencias o niveles, estos tres niveles representan los tres workflows principales disponibles en Scribe, que son Mark, Transcribe y Verify.

- **Groups:** Un grupo organiza los subjets en conjuntos relacionados. Por lo general los grupos se crean basados en similitudes de contenido o formato de subjects. Esto es muy útil para que el creador del proyecto pueda organizar, mantener y presentar los proyectos o colecciones de subjects.

- Una colección de diarios se puede agrupar en el Diario de Ann de 2012, el Diario de Bob de 2013 y el Diario de Cathy de 2014.
- Un libro de cuentas puede agruparse en páginas de índice, sección principal y apéndice.

⁸<http://scribeproject.github.io/>

⁹<http://whaling.oldweather.org>

¹⁰<http://measuringtheanzacs.org>

¹¹<http://emigrantcity.nypl.org>

- **Classifications:** Una classification es un registro de una respuesta de una tarea realizada por un usuario sobre un Subject. Algunos ejemplos de una classification son:
 - Una marca realizada por el usuario donde una fecha este escrita en una pagina.
 - Un usuario indica si una persona está presente o no en una fotografía.
 - Una transcripción de una marca realizada.

- **Workflows:** Un Workflow representa una serie de tareas que un usuario debe realizar sobre un conjunto de subjects. A continuación se detallara un ejemplo de un workflow para extraer fechas de un diario.
 - El usuario marca donde se encuentre escrita una fecha en una pagina de un diario.
 - El usuario transcribe la fecha que fue marcada.
 - Otro usuario verifica que la fecha es correcta.

- **Tasks:** Una tarea es una acción que se solicita a los usuarios realizar sobre los subjects.
 - El usuario marca en la pagina de un diario si se encuentra una firma.
 - El usuario indica si existe o no una persona en una fotografía.

5.3.2. Configuración de un proyecto Scribe

Para configurar un proyecto Scribe debemos seguir una serie de pasos, a continuación se detallara los pasos mínimos necesarios para poder crear un proyecto con el framework Scribe.

Lo primero es descargar el código fuente desde el repositorio. Una vez descargado el código fuente, se debe crear una estructura de carpetas, cuya carpeta raíz debe tener el nombre del proyecto. La estructura debe ser como se muestra en la figura [5.8](#).


```
my_project/  
+-- content/  
|   +-- help/  
+-- assets/  
|   +-- css/  
|   +-- images/  
|   +-- javascript/  
+-- subjects/  
+-- tutorial/  
+-- workflows/  
+-- project.json
```

Figura 5.8: Estructura proyecto Scribe

En el archivo `project.json` se define la identidad del proyecto, como el título, la descripción, detalles de la organización, logo del proyecto entre otras configuraciones.

Configuración de Workflows

Un workflow representa la serie de tareas de marcado, transcripción o verificación, para un conjunto de subjects (por ejemplo, documentos, imágenes). Este es el paso más importante y completo para configurar el proyecto.

La configuración los workflow se realiza dentro la siguiente estructura [5.9](#) y por cada workflow se definen en un archivo [5.10](#) de configuración con extensión `.json`.

```
my_project/  
+-- workflows/  
|   +-- mark.json  
|   +-- transcribe.json  
|   +-- verify.json
```

Figura 5.9: Estructura workflow Scribe

```

{
  "name": "mark",
  "label": "Mark Workflow",
  "retire_limit": 3,
  "generates_subjects": true,
  "generates_subjects_for": "transcribe",
  "first_task": "the_first_task",
  "tasks": {
    "the_first_task": {
      "next_task": "the_second_task"
    },
    "the_second_task": {
      "next_task": "the_last_task"
    },
    "the_last_task": {
      "next_task": null
    }
  }
}

```

Figura 5.10: Estructura archivo de un workflow

Dentro del archivo de configuración del workflow se definen los siguientes parámetros:

- **name:** El nombre único que identifica el workflow.
- **label:** Se define un nombre “amigable” para representar el workflow.
- **generates_subjects:** Indica si el workflow genera subjects.
- **generates_subjects_for:** Indica el nombre del siguiente workflow, por ejemplo, para un workflow de transcribir, el siguiente puede ser verificar.
- **first_task:** Indica la tarea inicial del workflow.
- **task:** son acciones que se realizan sobre un subject, cada workflow puede tener múltiples task, que pueden o no estar ordenadas.

Dentro del archivo de configuración del workflow también se define la configuración de una task [5.11](#), esta configuración consta de:

- **key:** Indica el el nombre que la identifica
- **instruction:** Una instrucción de lo que el usuario debe realizar.
- **next_task:** contiene la key de la siguiente tarea a realizar.
- **tool:** Scribe tiene predefinidos algunos widgets que ayudan a realizar la task, una tool puede ser un cuadro de texto para que el usuario lo complete, una herramienta de marcado o un dialogo para seleccionar entre múltiples opciones.

```
tasks: {
  "determine_has_animals": {
    "tool": "pickOne",
    "instruction": "Are there animals in this image?",
    "tool_config": {
      "options": {
        "yes": {
          "label": "Yes",
          "next_task": "identify_animals"
        },
        "no": {
          "label": "No"
        }
      }
    }
  }
}
```

Figura 5.11: Estructura de configuración de una task

Configuración de subjects

Un Subject es un documento multimedia identificable de forma única, que normalmente contiene una imagen, en el que se pide a los usuarios que trabajen.

Los subjects [5.12](#) se definen en un archivo .csv y deben ubicarse dentro del directorio /subjects.

cada archivo csv representan un conjunto de subjects, y tiene la siguiente estructura definida:

- **order:** Define la secuencia de los subjects.

- **file_path:** La URL del archivo multimedia completo.
- **thumbnail:** Indica la URL de la imagen en miniatura del archivo multimedia.
- **width:** Indica el ancho en píxeles del archivo multimedia.
- **height:** Indica la altura en píxeles del archivo multimedia.
- **set_key:** Indica si pertenece a un grupo.

```
key,name,description,cover_image_url,external_url,meta_data_1,retire_count
dogs,The Dog Group,A Collection of Dog Images,http://placepuppy.com/400/300,http://en.wikipedia.org/wiki/Dog,Doggy dat
cats,The Cat Group,A Collection of Cat Images,http://placekitten.com/400/300,http://en.wikipedia.org/wiki/Cat,Kitty Da
mice,The Mouse Group,A Collection of Mouse Images,http://placemouse.com/400/300,http://en.wikipedia.org/wiki/Mouse,Mic
...
```

Figura 5.12: Ejemplo de archivo de subjects.

Configuración de Groups

Los grupos se definen en un archivo llamado groups.csv indicando los siguientes campos:

- **key:** Es el nombre en el que refiere al nombre de archivo de subjects definidos. Por ejemplo dogs se refiere a un conjunto de subjects group_dog.csv
- **name:** Indica el nombre del grupo que se muestra.
- **description:** Indica la descripción del grupo.
- **cover_image_url:** Indica la url de una imagen que representa el grupo

5.4. Anotación semántica de textos: Recogito

Recogito¹² es una plataforma de documentos colaborativos, mantenida por Pelagios (una iniciativa de Humanidades Digitales), la cual ofrece un espacio personal de trabajo donde se puede cargar, recopilar y organizar materiales fuentes (textos, imágenes y datos tabulados) y colaborar con anotaciones e interpretaciones, que pueden representar texto, un lugar, un evento o una persona y poder exportarlos en distintos formatos como lo son el formato CSV, RDF y JSONLD.

En la plataforma hay un propietario, el cual se encarga de subir los documentos, ya sea en formato de texto (archivo txt), formato de imagen o un csv (datos tabulados) y además puede asignar colaboradores para permitir que otros usuarios puedan ayudar en la creación de anotaciones en los documentos.

Anotaciones

La anotación de Recogito es la asignación de un valor a una selección identificada en el documento. Existen dos formas de selección y a su vez existen 4 tipos de anotaciones.

¹²<https://recogito.pelagios.org/>

Las maneras de selección dependen del tipo de documento (texto o documento) que se quiera agregar anotaciones. Si es de texto o es una imagen, se proporciona una herramienta de selección en toda la imagen, esta puede ser una selección de forma rectangular, un punto o cuadro inclinado. Para la selección de texto solo hace falta hacer una selección con el mouse del texto a anotar, como en cualquier editor de texto.

Las anotaciones pueden ser de 4 tipos:

- **Texto (Text):** Sólo se agrega un comentario o etiquetas, para representar una descripción de la selección.

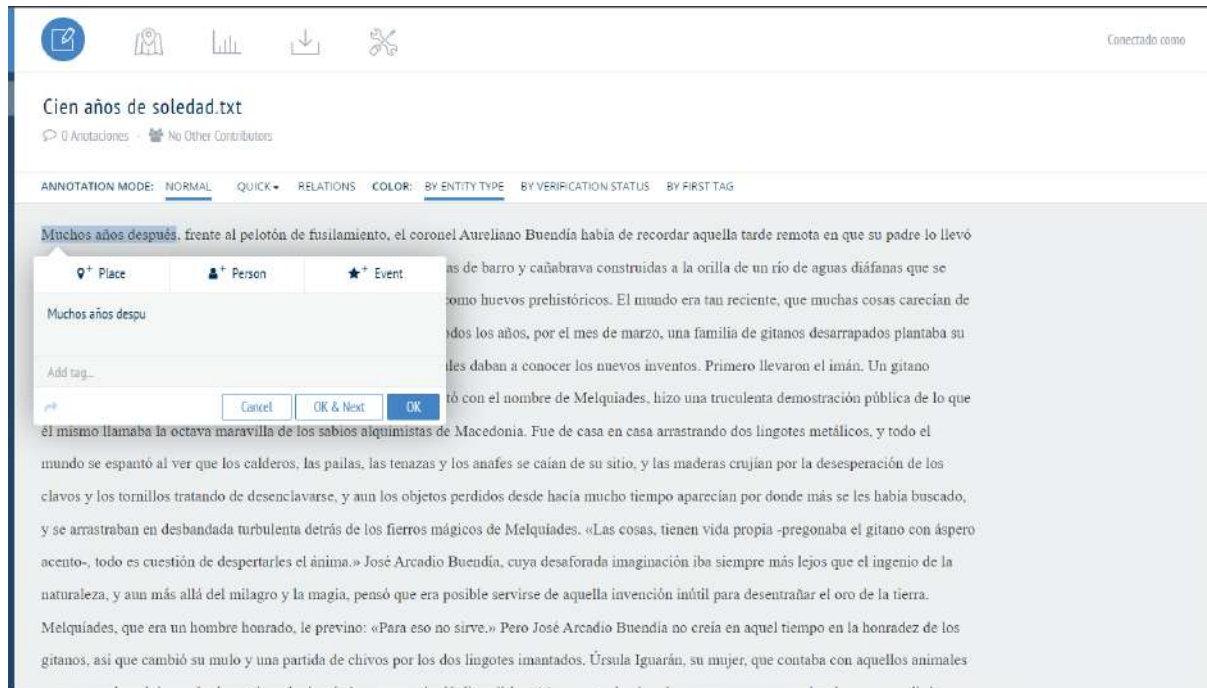


Figura 5.13: Ejemplo de creación de una anotación de texto

- **Lugar (Place):** Para ingresar un lugar, la plataforma muestra un mapa y hace falta hacer la búsqueda del lugar correspondiente. Para acelerar la búsqueda de un lugar, al seleccionar la opción lugar, automáticamente se busca el texto seleccionado.



Figura 5.14: Ejemplo de anotación de lugar.

- **Persona (Person):** Solo se marca la anotación como una persona.
- **Evento (Event):** Solo se marca como un evento.

5.5. Conclusiones

En este capítulo se han detallado diversas herramientas para la transcripción de documentos manuscritos. Si bien el proyecto FromThePage se adapta mejor a los objetivos de esta tesis, en general, estas herramientas carecen de funcionalidades para relacionar documentos, ofrecer alternativas de transcripción o representar objetos del mundo real en la transcripción.

Recogito, por ejemplo, permite la representación de objetos además de texto, pero se limita únicamente a la ubicación, eventos o personas, y estos objetos no pueden utilizarse para representarse en otros documentos.

Con el desarrollo de Transcriptor, se busca solucionar estos aspectos, ofreciendo las herramientas necesarias para lograr transcripciones, relacionar y buscar documentos, y proporcionar un vocabulario para representar objetos del mundo real tales como sellos, autores y firmas.

Parte II

Diseño e Implementación

Capítulo 6

Estrategia general

En este capítulo, se presentará el desafío y el diseño de la solución que enmarca esta tesis, cuales son los objetivos de negocio a los que se aspira a resolver y cuales son las características principales de la solución propuesta. Se analizarán cuales son los problemas y mejoras sobre los que se deben trabajar, para luego planificar cómo solucionarlos.

Como se mencionó en el capítulo 2 de esta tesis, transcribir un manuscrito es un trabajo que conlleva mucho esfuerzo y desgaste intelectual. Esto se debe a que los involucrados en estos proyectos requieren interpretar constantemente el texto para poder extraer su contenido y volcarlo en otro medio. Con este panorama, el surgimiento y utilización de herramientas de trabajo colaborativo dieron lugar a un nuevo enfoque, mejorando y agilizando el proceso e introduciendo a la ciencia ciudadana como pilar en estas investigaciones comunitarias. Este enfoque, en donde ya la comunidad no está formada exclusivamente por investigadores profesionales, sino que también por aficionados e interesados, necesita de disponer un espacio propicio donde se pueda organizar el trabajo de transcribir así como también disponer de las herramientas necesarias para llevarlas a cabo. En este contexto también aparecen ciertos planteamientos tales como:

- Permitir la creación de proyectos de transcripción a cualquier miembro de la comunidad: Cualquier usuario puede tener la oportunidad de crear un proyecto de transcripción, para que cualquier otro usuario de la comunidad pueda participar y colaborar.
- Facilitar la incorporación de metadatos a los elementos presentes en el manuscrito digitalizado, sean de naturaleza textual o pictórica, para que los mismos puedan ordenarse, categorizarse y buscarse de forma automática.
- Proveer un método de curado de la información para asegurar la calidad de la misma.
- Mantener el interés y alentar a que los participantes colaboren en los proyectos.
- Organizar y presentar la información recabada.

6.1. Introducción

Considerando las problemáticas y desafíos inherentes a la interpretación y creación colectiva de datos, se concibió un conjunto de estrategias diseñadas con el propósito de ofrecer soluciones pertinentes para cada uno de estos aspectos. El producto final, al cual se lo denominó Transcriptor, tiene como objetivo principal fomentar la participación activa de la sociedad en la interpretación de manuscritos antiguos, asignando a cada individuo un

papel significativo en dicho proceso. Esto posibilita que cada participante aporte su perspectiva y conocimiento personal acerca del mensaje del documento, generando así una dinámica enriquecedora que contribuye al ejercicio de la ciencia ciudadana. A continuación, se detallarán estas estrategias, presentándolas como características que la plataforma incorporará, con el fin de facilitar y potenciar la experiencia del proceso de transcripción.

6.2. Características

6.2.1. Usuarios

Los usuarios conforman un rol fundamental en la plataforma, son los encargados de proveer, transformar y consultar la información de los manuscritos, interactuando entre sí para realizar un curado colaborativo de los datos ingresados. Entre ellos podemos definir varios perfiles que están diferenciados por la labor que desempeñan dentro de la comunidad.

- **Transcriptor:** Es el encargado de realizar el acto de transcripción, su objetivo es aportar su concepción visual del manuscrito y de esta manera identificar y clasificar las marcas de los mismos para luego traspasarlas a formato digital de la manera más cercana posible al original digitalizado. A su vez puede participar de foros de discusión sobre fragmentos transcritos por otros usuarios y votar por las mejores versiones de las mismas.
- **Administrador de proyecto:** Es el creador de un proyecto de transcripción, el cual administra la configuración del mismo, puede modificar las páginas cargadas, editar los detalles de un trabajo, definir reglas de transcripción y funcionar como moderador ante situaciones que surjan entre con los documentos o los transcriptores durante el proceso. Tiene acceso además a visualizar las estadísticas e historial de progreso de su proyecto.
- **Administrador:** Tiene acceso a las estadísticas de la comunidad y utilización de la plataforma, puede visualizar el historial de acciones para todos los proyectos. Puede realizar modificaciones sobre parámetros de configuración y editar la información de cualquier proyecto.

6.2.2. Transcripciones

Las transcripciones son la característica central que nuclea a todas las entidades de la plataforma, estas van a ser quienes representen el contenido procesable de los documentos digitalizados y las mismas pueden mejorarse progresivamente a través de la colaboración de los usuarios para ajustarse más a una representación fiel del original. Como uno de los objetivos de este trabajo es mantener la mayor información de manuscritos digitalizados en formato procesable e indexable, se planteó un modelo de transcripción en capas donde cada una va a describir distintos aspectos del documento buscando siempre alcanzar el máximo nivel de detalle en cada uno, para no solo mantener el contenido textual del mismo, sino generar una gran descripción de todos los niveles perceptibles del documento.

6.2.3. Capas de Transcripción

Las capas son la representación de características de un manuscrito agrupadas bajo un criterio, esto nos permite separar en niveles los detalles que queremos transcribir de

un manuscrito y de esta manera no mezclar conceptos como por ejemplo: el texto escrito en el cuerpo de una página y las notas al margen de la misma. En esta etapa podemos diferenciar cuatro tipos de capa:

- **Capa principal de texto:** Representa el contenido principal del documento, lo que está escrito en el cuerpo del manuscrito y lo primero que se tiende a leer.
- **Capa Secundaria de texto:** Utilizada para mantener el contenido textual secundario del documento, encabezados, notas al margen, correcciones y pies de página, etc.
- **Capa de ilustraciones:** Agrupa la representación de las imágenes, fotos y figuras que se pueden observar.
- **Capa de sellos y marcas registradas:** Representa el contenido de sellos y marcas de impresión que pueda contener el documento.
- **Capa de detalles:** Reservada para contener descripciones de marcas o accidentes del paso del tiempo o de algún contexto que valga la pena aclarar.

6.2.4. Capas de transcripción semántica

Durante la transcripción de un documento, se realiza la lectura y análisis de las palabras para su interpretación y transcripción. Esta traducción literal de los símbolos permite al soporte destino contar con la información principal que el original conserva. A su vez, existen otros elementos que a menudo no suelen ser textuales y que guardan otros datos que pueden ser relevantes para el contexto del manuscrito. En función a estos objetos de valor pictórico como pueden ser firmas, imágenes y sellos, pueden distinguirse distintos niveles de transcripción para una página. Estos niveles se podrían dividir en “literal” y “descriptivo” o llevado a términos prácticos “textual” y “gráfico”.

Cabe destacar que la transcripción textual de un manuscrito suele ser una labor más sencilla que una transcripción descriptiva. Y esto se debe a que el objetivo del transcriptor es transferir la información de un soporte físico a un esquema digital, que cuenta con beneficios como la indexación y la reutilización pero también limita a que los datos estén construido únicamente con caracteres alfanuméricos.

En este contexto, omitir los elementos no textuales presentes en los documentos involucraría una potencial pérdida de información lo cual genera resultados incompletos que, según los casos, pueden privar a la transcripción de validez, contexto y autenticidad.

Una vez comprendida la importancia de estos objetos, el desafío planteado es generar un mecanismo eficaz para representarlos y garantizar un mayor nivel de fidelidad en cuanto al manuscrito original.

Como se mencionó en el apartado anterior, al utilizar un sistema de transcripción por capas es posible encapsular elementos de la transcripción bajo distintos criterios, como puede ser una sección de texto, otra para firmas, capas para almacenar lugares que menciona el documento, etc. Dentro de cada capa es posible que se requiera transcribir o describir elementos distintos, que no solo sean títulos o párrafos, sino conceptos o figuras dentro del manuscrito.

Para esto es necesario que las capas mantengan un registro no solamente textual de la información, sino que también guarden datos extra que ayuden a darle una descripción más adecuada. En función a la estructura que conlleva la definición de estas capas se plantearon las denominadas “capas de transcripción semántica”.

Las “transcripciones semánticas” son transcripciones con información adicional (metadatos) que estructura y aumenta el significado del aporte y además permite relacionar

con otros conceptos existentes dentro de Transcriptor. Este concepto incorpora una herramienta más para poder representar distintos elementos que componen un documento manuscrito, permitiendo mantener y/o aumentar la expresividad de la información resultante manteniendo los beneficios del contenido guardado en formato digital.

Aunque esta tesis no trata sobre conceptos específicos de la web semántica, se hacen uso de estos para lograr su cometido, los cuales se abordaron a modo informativo en el apartado 3.

6.2.5. Marcas

Según lo abordado con anterioridad, las transcripciones están conformadas por capas, las cuales nos permiten representar diversos aspectos del documento. A su vez, estas capas están formadas por referencias denominadas "Marcas", estas marcas contienen atributos posicionales, los cuales sirven para identificar la posición dentro del documento digitalizado donde se encuentra cierto elemento a transcribir y un atributo que representa el valor del contenido donde se guarda el texto plano del elemento transcripto por el usuario.

6.2.6. Método de transcripción

Los elementos presentados hasta el momento permiten elaborar un primer flujo de transcripción. Un usuario con perfil "Administrador de proyecto" genera un nuevo proyecto de transcripción y carga los originales en digital en la plataforma, a partir de este momento, cualquier usuario "Transcriptor" podrá comenzar un proceso de transcripción sobre los documentos subidos. El transcriptor ingresa al apartado de transcripción de la página que desee y visualizará una pantalla de dos columnas, donde de un lado podrá ver la imagen digitalizada de la página y en el otro un campo de texto donde transcribirá su contenido, por defecto la capa principal de texto estará habilitada. Opcionalmente el transcriptor puede crear una nueva capa de transcripción, seleccionando la opción "Nueva Capa" e incluyendo una denominación y eligiendo el tipo de capa a crear (Texto Secundario, Ilustraciones, Sellos y Marcas Registradas, Detalles). A partir de este punto el usuario tendrá dos formas de generar su primera contribución a la transcripción:

- Marcar en la imagen la línea que desea transcribir y volcar el contenido textual en el campo de texto
- Ingresar el contenido en el campo de texto y luego marcar a qué línea del documento corresponde.

Una vez realizada, la contribución esta habilitada poder ser votada y comentada por el resto de la comunidad. A su vez otros usuarios podrán sugerir sus propias interpretaciones del texto marcado y "competir" con votos para convertirse en la transcripción definitiva de cierto fragmento del documento.

6.2.7. Gamificación

En esta tesis se aplicarán dinámicas de gamificación con el objetivo de motivar la colaboración y contribución de la comunidad en los proyectos de transcripción. Para esto se diseñaron pequeños metajuegos que hacen más entretenido y ameno el proceso de transcripción al usuario, los cuales pueden apreciarse en múltiples flujos dentro del sistema.

Rangos e Insignias

Una de las técnicas utilizadas involucra logros e insignias, los cuales son otorgados a los usuarios a través del tiempo y las acciones realizadas dentro de la plataforma, dándole la posibilidad de escalar a través de una jerarquía de rangos. Este flujo y lógica de negocio fue implementada utilizando el framework Metagame [21], el cual vamos a ahondar en capítulos posteriores. Las insignias que el usuario va obteniendo se dividen en tipos, entre definidas podemos nombrar:

- Insignias de Inicio de Sesión
 - Bienvenido a Transcriptor (I Was Here)
 - Segundo Login (Welcome Back!)
- Insignias de Contribución
 - Primera contribución
 - Segunda contribución
- Insignias Especiales
 - Primera colección creada
 - Primer proyecto creado
 - Primera subida de archivos
 - Primera mejora de transcripción
- Insignias de Refuerzo
 - Primer refuerzo
 - Segundo refuerzo

La jerarquía de rangos definida es la siguiente, en orden de obtención:

- Visitor
- Explorer
- Citizen Scientist
- Prolific Citizen Scientist
- Committed Citizen Scientist
- Visionary

6.2.8. Votación de Marcas

Otro de los flujos que implementan técnicas de gamificación es la competencia por la transcripción definitiva de un fragmento marcado dentro del manuscrito. Los usuarios propondrán transcripciones alternativas a la realizada originalmente y estas se someten a votación a través del mecanismo de Likes. La transcripción con mayor valoración será la considerada como la más acertada según la comunidad y será la que aparezca en primera instancia en el contenido final transcripto.

Capítulo 7

Arquitectura e Implementación

7.1. Introducción

A continuación se describirá en detalle las decisiones de diseño y las tecnologías utilizadas para implementar la arquitectura de Transcriptor, cabe destacar que la implementación de Transcriptor es de código abierto, es decir, el código fuente es accesible, el cual mantiene una licencia Affero GPL y se encuentra documentado de manera pública en los siguientes repositorios en la plataforma Github:

- Transcriptor backend: <https://github.com/cientopolis/transcriptor-backend>
- Transcriptor frontend: <https://github.com/cientopolis/transcriptor-web>

7.2. FromThePage

FromThePage, presentado en el capítulo 5 de esta tesis, es una aplicación web de transcripción de manuscritos colaborativos de código abierto y extensible que cuenta con la licencia Affero GPL. El código fuente se encuentra publicado y puede obtenerse en el siguiente repositorio público Github:

- FromThePage: <https://github.com/benwbrum/fromthepage>

Como se mencionó, FromThePage cuenta con importantes características que permiten crear un buen marco de trabajo para los proyectos colaborativos de transcripción, englobando tanto la subida de documentos como el proceso de escritura y transcripción de contenido.

FromThePage esta construido sobre el framework RubyOnRails, el cual plantea una arquitectura MVC clásica, donde todas las vistas son renderizadas por el servidor, el procesamiento de texto esta delegado a los controladores y el modelo plantea la persistencia de datos y comportamientos puntuales dentro del modelo de negocio. Sumado a estas características, el hecho de que el proyecto sea de código abierto brinda facilidad de mantenimiento y extensibilidad en sus funcionalidades. Estos puntos coinciden con los objetivos base que plantea Transcriptor por lo que usaremos FromThePage como producto base sobre el cual construir la solución final.

7.3. Arquitectura General

Transcriptor emplea un modelo Cliente-Servidor, donde el servidor es el encargado de proveer los recursos y servicios necesarios para el procesamiento de la información enviada por el cliente.

En el servidor se alojará la aplicación Núcleo (Core o Backend) de la solución que expondrá sus servicios a través de una API Rest, la cual se encargará de proveer una interfaz apta para el envío/consumo de los datos entre las distintas aplicaciones clientes.

A su vez, Transcriptor cuenta con una aplicación cliente (Web o Frontend), la cual cumplirá con la función de enviar al Backend información provista por el usuario y mostrar los resultados obtenidos de los servicios (ver figura 7.1).

Transcriptor Architecture

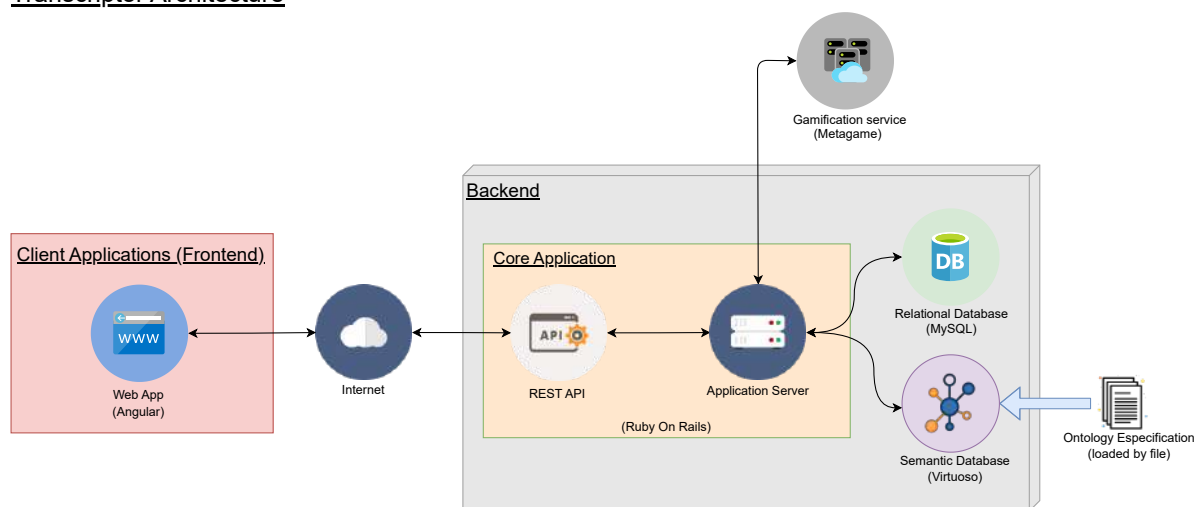


Figura 7.1: Diagrama de la arquitectura general de Transcriptor, la cual plantea la integración con múltiples clientes a través de una API Rest.

De esta manera, este modelo provee un sistema de integración lo suficientemente flexible como para ofrecer servicios tanto a un sitio web como a una aplicación móvil, bots o otras redes como Cientopolis, centralizando la lógica en un solo proyecto y delegando la complejidad de la presentación de los datos a cada aplicación cliente. Utilizando un enfoque más descriptivo, esta arquitectura nos brinda ventajas como:

- **Proveer separación cliente/servidor**

Al ser sistemas independientes (solo se comunican con un lenguaje de intercambio como JSON) es posible desarrollarlos como proyectos autónomos, equipos autónomos. De esta manera si es necesario evolucionar/refactorizar uno de los dos, cliente o servidor, se puede hacer de manera separada, siempre que se mantenga la interfaz de la API. Esto permite a su vez alta escalabilidad en cuanto a clientes permitiendo el desarrollo de varios de ellos con un único servidor.

- **Independencia de tecnologías/lenguajes**

El desarrollo puede darse en cualquier tipo de tecnología o lenguaje según las especificaciones, tiempos de desarrollo, filosofía o necesidades del proyecto. También es indistinto que en el futuro se cambie totalmente las tecnologías con las que implementó el cliente o servidor, siempre y cuando se respete el contrato entre ellos,

esto es, que se siga teniendo las mismas operaciones en la API y se realice la misma funcionalidad que se supone que deben hacer.

- **Fiabilidad, escalabilidad, flexibilidad**

Se habla de fiabilidad porque mientras el nexo entre el cliente/servidor se respete y no haya cambios estructurales que impliquen un rediseño de esta interfaz, se pueden realizar modificaciones en ambas partes: lenguajes, bases de datos, etc. Todo esto siempre y cuando se retornen correctamente los datos necesarios de cada operación. Escalabilidad porque el crecimiento puede darse en cualquier momento. Los métodos de la API en el backend pueden responder a otros tipos de operaciones y hasta puede versionarse para dar lugar a nuevas funciones conservando su funcionalidad actual. El desarrollo del lado del cliente también puede escalar todo lo necesario con el tiempo, no solo como proyecto individual, sino que también en la implementación de otros frontends, ya sea web, aplicaciones móviles de distintas plataformas u otro tipo de integraciones.

A continuación haremos énfasis en el rol que cumple cada componente arquitectónico de la solución y en la descripción de sus partes.

7.4. Aplicación Core (transcriptor-backend)

La aplicación core o backend (a partir de ahora Transcriptor-Backend) es la capa encargada del procesamiento y respuesta de las solicitudes de todos los usuarios a través de una API REST. Es aquí donde se implementan los servicios necesarios para llevar adelante la lógica de negocio de Transcriptor, como lo es gestionar los documentos digitalizados, flujos de transcripción, manejo de usuarios y la comunicación con otros proyectos.

7.4.1. Tecnologías Utilizadas/Implementación

Para el Backend de Transcriptor se utilizó en su implementación

- **Ruby On Rails:** también llamado “Rails” es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby que incluye todo lo necesario para crear aplicaciones con el esquema “base de datos-backend” (database-backend) siguiendo el paradigma del patrón Modelo Vista Controlador (MVC).
- **Rails::API:** es un subconjunto de funciones de una aplicación Rails normal, creada para aplicaciones que no requieren toda la funcionalidad que proporciona una aplicación Rails completa, permite generar APIs más livianas y, en consecuencia, más rápidas que una aplicación Rails normal. El ejemplo principal para su uso es en backends API, donde generalmente no se necesita la pila completa de middleware Rails, ni la generación de plantillas de vista. Devise: Propone una solución modular de autenticación basada en el modelo MVC de Rails, brindando funciones de inicio de sesión por múltiples vías (credenciales, token, OAuth), gestión de registro y operaciones de claves de usuarios.
- **RMagick:** provee una interfaz entre el lenguaje de programación Ruby y la librería de procesamiento de imágenes ImageMagick, en Transcriptor es usada para ajustar la calidad de las imágenes para controlar su tamaño y para la generación de las vistas previas de cada página.

- **Metagame:** Sistema de Metajuego, el cual a través del registro de actividades en la aplicación establece puntajes y asigna medallas y rangos a los usuarios que las realizan
- **Virtuoso Universal Server:** Middleware y motor de base de datos que combina en un solo sistema las funcionalidades de un manejador de base de datos relacional y orientada a objetos, base de datos virtual, RDF, XML, free-text, servidor web y servidor de archivos.
- **Ruby RDF:** conjunto de gemas que conforman una librería modular que proveen funciones para facilitar el manejo y procesamiento de para elementos de datos enlazados y web semántica en Ruby.

7.4.2. Modelo

El Backend de Transcriptor constituye el componente principal de la solución, encargado de la lógica de negocio y el soporte en el cual se construye la API que conecta los servicios con los clientes finales. A continuación presentamos el diagrama UML con las principales clases que integran el sistema.

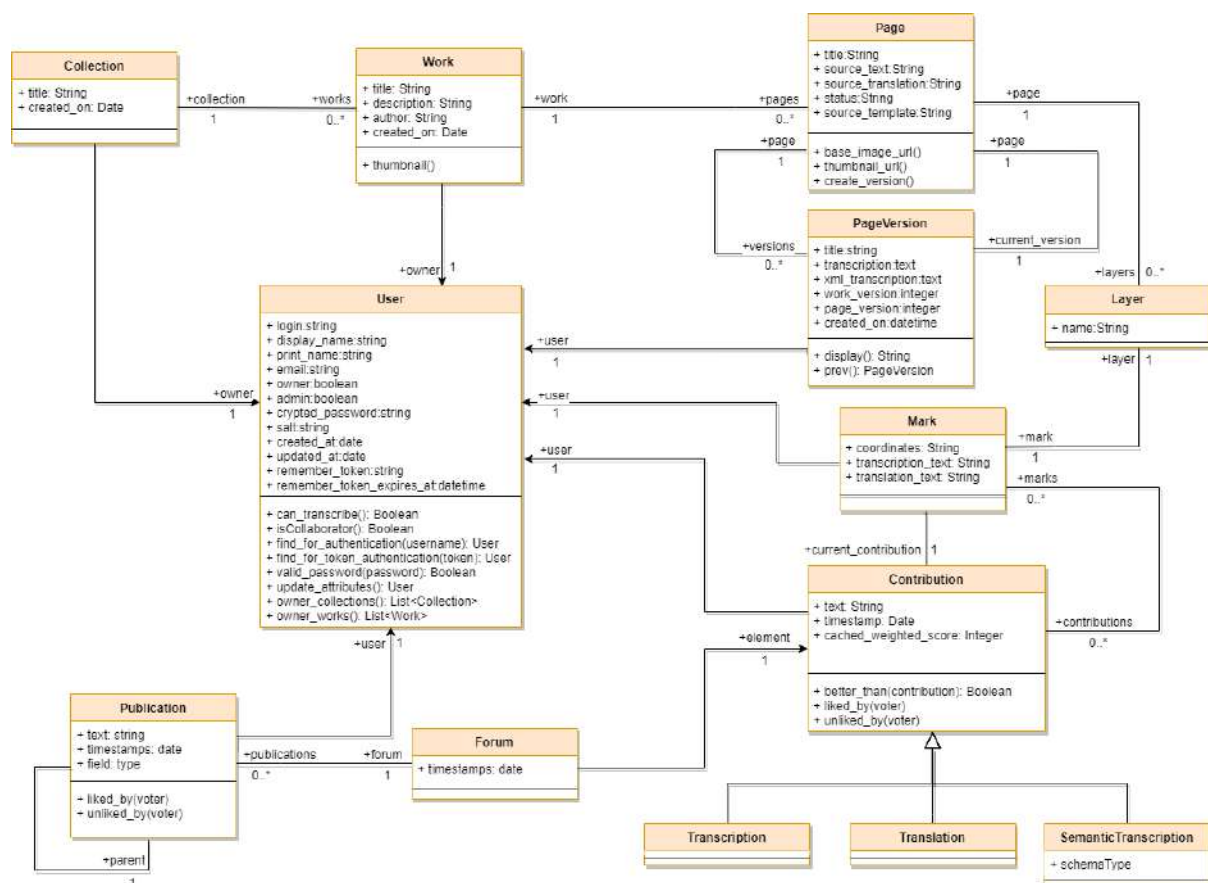


Figura 7.2: Diagrama UML de la estructura de clases principal de Transcriptor

Se procederá a describir brevemente cada uno de los componentes y funcionalidad representados en el diagrama de clases, para que una vez identificados podamos explicar como se relacionan e interactúan entre si en las distintas funciones de la API.

- **User (Usuario):** Objeto que representa a un usuario de Transcriptor, posee atributos de identificación tales como el nombre de usuario (login) y el email los cuales los utilizara al momento de iniciar sesión en la plataforma.
- **Collection (Colección):** Agrupa proyectos de transcripción, los cuales están relacionados bajo algún contexto (mismo autor, época, estilo, etc). Esta identificado bajo un titulo, elegido por el usuario que genera la misma.
- **Work (Proyecto):** Representa un proyecto de transcripción, mantiene el conjunto de paginas las cuales serán transcritas. Los Works, así como las Collections se identifican con un titulo y opcional pueden contener información sobre el conjunto de paginas en si, como lo son el autor, descripción e historia del documento, año de confección/publicación.
- **Page (Página):** Clase que describe los atributos y características de las paginas a transcribir, almacena el texto en formato legible y en formato XML permitiendo que el transcriptor pueda incluir tags que describen elementos del manuscrito. Además mantiene las referencias a la imagen del documento digitalizado y contiene el estado en el el cual se encuentra la transcripción que pueden variar entre “Página Vacía”, “Necesita revisión”, “Pendiente” o “Transcripto”.
- **Layer (Capa):** Asocia un conjunto de marcas de transcripción realizadas sobre un documento, Transcriptor soporta dos tipos de capas: “Capa de Texto” y “Capa Semántica”
- **Mark (Marca o Anotación):** Representa un polígono marcado sobre un área del documento a la cual se le asocia la transcripción de la sección referenciada.
- **Contribution (Contribución):** Objeto que describe el texto propuesto por un transcriptor para una sección del manuscrito. Las Contributions pueden ser de tres tipos:
 - *Transcription (Transcripción):* texto que alude al contenido textual del fragmento seleccionado.
 - *Translation (Traducción):* texto que hace referencia a la traducción de la sección marcada a un idioma particular.
 - *SemanticContribution (Contribución Semántica):* representa una descripción semántica del fragmento, la misma puede estar realizada utilizando una colección de tipos los que presentaremos más adelante.

Adicionalmente todas las contribuciones implementan la interfaz Votable, lo que permite que los usuarios puedan votar por aquellas que consideren que reflejan mejor el contenido de cada sección del documento.

- **PageVersion (Versión de Pagina):** Representa una versión de una pagina, esto es de utilidad para mantener el registro de los cambios que se realizaron en los textos de la misma. Al momento de realizar la edición sobre el texto transcripto de una pagina, se crea una nueva instancia de PageVersion la cual pasa a referenciarse como la versión actual (propiedad `current_version`) de esta.
- **Forum (Foro):** Foro de contribuciones, objeto que describe un canal de comunicación basado en un objeto, en este caso de una contribución, y agrupa los comentarios de los usuarios.

- **Publication (Comentario):** describe una publicación en el Forum, puede representar tanto a un comentario de primer nivel como a la respuesta a otro mensaje de la cadena. Implementa el protocolo Votable para incluir la dinámica de votación (“Me Gusta” o like).

7.4.3. Modelo de interacción

Una vez presentados los componentes de la solución, podemos describir un primer modelo en el cual los objetos interactúan para lograr el objetivo de la plataforma.

Transcripción

El componente inicial sobre el cual se centra este flujo es la Page (página), esta contiene toda la información relacionada a una página dentro un proyecto de transcripción (Work). Las instancias de Pages poseen además un atributo de tipo texto, la cual representa ni más ni menos, a la transcripción en texto plano de la misma, que se genera a través del proceso de marcado.

Todo parte de la generación de una anotación o marca (Mark) que contiene:

- La referencia al fragmento a transcribir del documento digitalizado, la cual está representada como coordenadas dentro de la imagen.
- Una asociación a una Transcription, la cual representa el texto del fragmento a transcribir.

Utilizando el texto de la marca creada es posible comenzar a generar la transcripción final en texto plano concatenando cada nuevo fragmento transcripto.

Mejora de transcripciones

Para fortalecer la colaboración dentro de la plataforma y mejorar los resultados, los usuarios pueden proponer transcripciones alternativas para cada fragmento del documento. Para lograr esto, cada anotación (Mark) realizadas sobre el documento tiene asociada a su vez una lista de Transcriptions secundarias las cuales pueden ser consultadas por cualquier usuario. A su vez, las Transcriptions alternativas también pueden llegar a ser la transcripción principal de la Mark, formando parte del texto final. Esto se consigue a través de un sistema de votación, donde cada Transcripción acumula un determinado número de vistos buenos (Likes o Me Gusta), por parte de los usuarios de la plataforma.

```

1  def like
2    @transcription = Transcription.find(params[:transcription_id])
3    @transcription.liked_by current_user
4
5    @mark=@transcription.mark
6    if(@transcription.better_than? @mark.transcription)
7      @mark.transcription=@transcription
8      @mark.save
9    end
10   record_deed(@transcription)
11   render_serialized ResponseWS.ok("api.contribution.transcription.like", @transcription)
12 end

```

Figura 7.3: Definición del servicio like, invocado para registrar un nuevo “Me Gusta” por parte de un usuario

En la figura [7.3](#) podemos ver la definición del servicio /like en la clase Transcription-Controller. En él se obtiene la Transcription persistida en la base de datos utilizando el transcription_id provisto en los parámetros de la solicitud y se registra el voto utilizando el método liked_by el cual recibe como parámetro el usuario actual(current_user).

```
1 def better_than?(another_contribution)
2   return self.id != another_contribution.id
3     && self.cached_weighted_score > another_contribution.cached_weighted_score
4 end
```

Figura 7.4: Definición del servicio better_than, utilizado durante el proceso de comparación para obtener la mejor transcripción entre las propuestas.

A continuación se realiza la invocación al método better_than de la clase Contribution el cual recibe como parámetro otra contribución (another_contribution), realiza la comparación por su cantidad de votos (cached_weighted_score) y retorna si esta tiene mayor cantidad, en caso positivo se actualiza la referencia de la Mark a la Transcription principal.

7.4.4. API (transcriptor-api)

Para proveer una comunicación con transcriptor-backend se diseñó una API (Application Programming Interface) de tipo REST (representational state transfer) denominada transcriptor-api, la cual provee una interfaz que permite a clientes ya sean web, aplicaciones mobile u otros proyectos a integrarse con Transcriptor y utilizar sus servicios. Esta interfaz se planteó en base a las acciones más importantes dentro de la plataforma, buscando la facilidad de integración brindando un conjunto de endpoints que priman la escalabilidad de la solución. En este inciso nos centraremos en inducir a las especificaciones de las solicitudes y respuestas aceptadas por la interfaz y los métodos de seguridad empleados para luego centrarnos en los principales métodos de la API describiendo sus flujos y respuestas.

Solicitudes

Al basarse en el estilo de arquitectura REST, la API de transcriptor, busca aprovechar y hacer uso de la semántica y funcionalidad de los métodos HTTP para realizar todas las acciones dentro del transcriptor-backend proporcionando un set de operaciones intuitivo para quien haga uso de la misma. Cada una de las solicitudes reciben a través de parámetros de URL los datos necesarios para realizar la operación en cuestión y eventualmente, en el caso de que sea necesario también se solicitan datos a través del cuerpo de la solicitud en formato JSON.

A modo de ejemplo podemos tomar como referencia el método GetWork que como sugiere su nombre en inglés, obtiene la información de un Proyecto(Work) anteriormente creado, utilizando su id para referenciarlo de la siguiente manera.

```
GET /api/work/{work_id}
#work_id es el id del proyecto en cuestión
```

Otro ejemplo de solicitud es el servicio *CreateCollection*, el cual genera una nueva Collection donde se podrán agregar proyectos para su transcripción, a diferencia del anterior,

podemos notar que este es llamado a través de el método HTTP POST y recibe en su cuerpo los datos de la nueva Coleccion(Collection) a crear.

```
1 POST /api/collection
2
3 {
4   "title": "Cuentos Argentinos",
5   "description": "Un compilado de manuscritos de viejos cuentos de
6     Argentina."
7 }
```

Figura 7.5: Ejemplo con datos aceptado por el servicio de creación de colecciones.

Respuestas

Para los datos de las respuestas transcriptor-api utiliza el formato JSON el cual sigue una estructura específica. Esto se debe a que los datos de la respuesta se encuentran encapsulados en un objeto genérico al que denominamos *ResponseWrapper* en el cual, además de contener los datos solicitados en la petición HTTP, contiene atributos del estado en el que la operación finalizó y mensajes para mostrar al usuario en el cliente que realizó la solicitud.

Tomando un ejemplo del inciso anterior, la respuesta al método *CreateCollection*, que se encarga de dar de alta una nueva colección de manuscritos en el sistema, sería la siguiente:

```
1 {
2   "status": "OK",
3   "message": "Collection has been created",
4   "data": {
5     "id": 6,
6     "title": "Cuentos Argentinos",
7     "owner_user_id": 2,
8     "created_on": "2020-03-29T22:27:28.914Z",
9     "intro_block": null,
10    "footer_block": null,
11    "restricted": false,
12    "picture": {
13      "url": null,
14      "thumb": {
15        "url": null
16      },
17      "scaled": {
18        "url": null
19      }
20    },
21    "supports_document_sets": false,
22    "subjects_disabled": false,
23    "transcription_conventions": "",
24    "slug": "cuentos-argentinos",
25    "review_workflow": false,
```

```

26     "hide_completed": true,
27     "help": "",
28     "link_help": ""
29 },
30 "alert": {
31     "title": "New Badge Obtained!",
32     "message": "First Collection"
33 }
34 }

```

En la respuesta podemos ver como en el primer nivel del JSON resultante tenemos las claves que involucran al ResponseWrapper:

- **status:** Indica el estado de manera resumida, el cual es útil comparar en condiciones, “OK” indica que la operación terminó en buenos términos mientras que “ERROR” indica que ocurrió un error en la transacción.
- **message:** Mensaje a mostrar al usuario, permite dar feedback del estado en el que acabó la operación
- **data:** El dato concreto en cuestión, en este caso es la colección creada a partir de los parámetros enviados en la figura [7.5](#).
- **alert:** Representan alertas de eventos secundarios que provocan la acción que acaba de realizar el usuario, en este caso la creación de esta colección hizo que el usuario gane la medalla “Primera colección”.

Controladores

Los controladores (*controllers*) son uno de los componentes fundamentales en la arquitectura MVC, los cuales se encargan de invocar peticiones y manipular el modelo para finalmente enviar los datos obtenidos a la vista, para realizar la presentación de los mismos al usuario. En el caso de Transcriptor la lógica es la misma, solo que el concepto de vista pasa a ser la interfaz planteada por la API. Los controladores reciben los datos enviados a través de las solicitudes HTTP desde la API, procesan la entrada y envían la consulta al modelo, el cual compone la lógica de negocio final, una vez resuelta la consulta los datos pasan nuevamente al controlador, el cual los procesa y genera una respuesta, la cual será enviada a través de la API. Transcriptor-API modela sus controladores utilizando una jerarquía, la cual permite generalizar las tareas comunes dentro de la API, facilitando el mantenimiento y el desarrollo de nuevas funcionalidades.

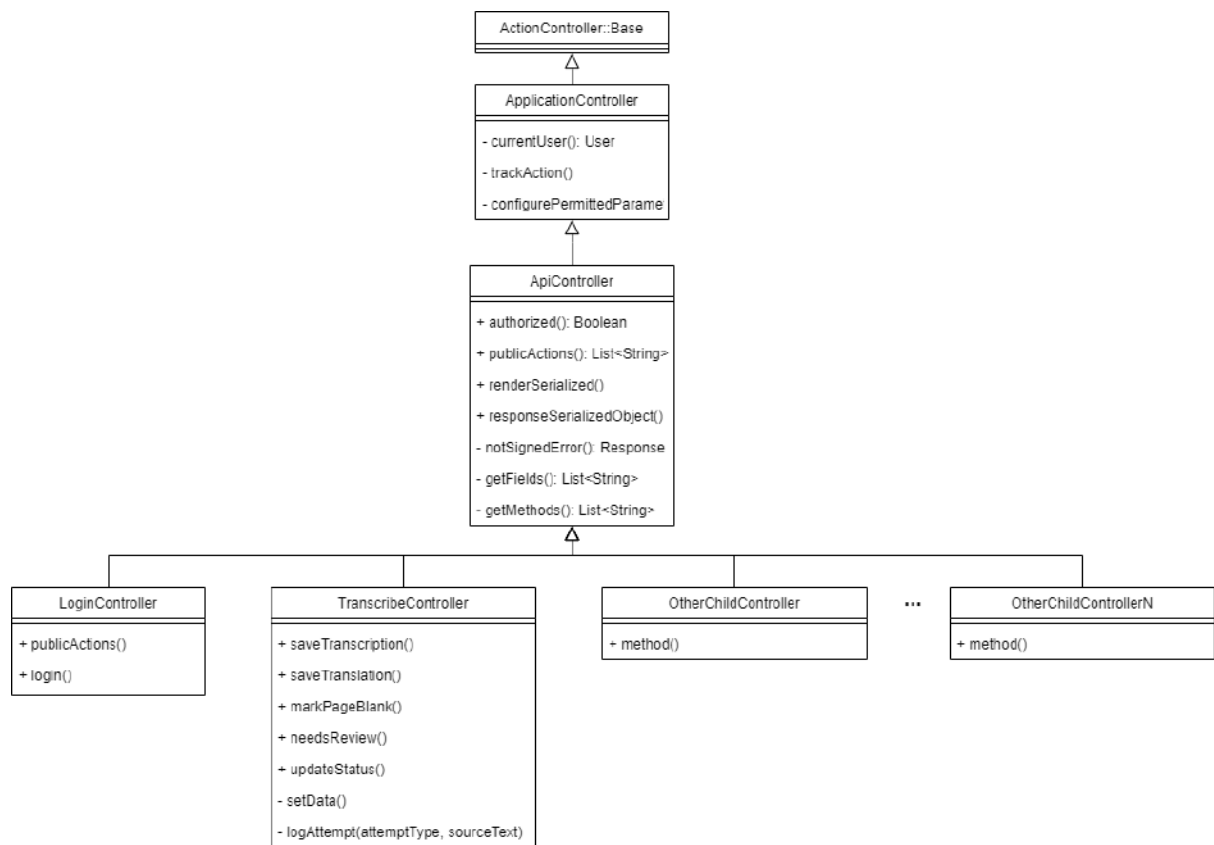


Figura 7.6: Jerarquía de controladores de la Transcriptor-API

Como se describe en la figura [7.6](#), la estructura de controladores esta planteada en una jerarquía, donde los controladores finales o hijos, son los que atienden los requerimientos a la API, a su vez existen métodos comunes a todos ellos, los cuales fueron generalizados en la clase *ApiController*, como ejemplo podemos tomar el método *responseSerializedObject* el cual recibe un objeto y se encarga de transformarlo en una respuesta de la API.

El *ApiController* también es hijo de la clase *ApplicationController*, el cual se encarga de proveer funciones para obtener la sesión del usuario que esta realizando el llamado y métodos de seguimiento de la actividad del mismo. Finalmente el *ApplicationController* es una subclase de *ActionController::Base*, la cual compone el núcleo de la lógica de solicitudes web en Rails, es decir, provee la implementación básica del controlador.

Seguridad

Transcriptor-API se desarrolló siguiendo las pautas de seguridad planteadas por la documentación de Ruby on Rails, las cuales enuncian las buenas prácticas sobre este tipo de aplicaciones, adicionalmente la gema Devise provee el soporte para generar Tokens de acceso que nos permiten autenticar al usuario en la plataforma.

El concepto general detrás de la autenticación basada en token es simple, permitir que los usuarios ingresen su nombre de usuario y contraseña para obtener un token, el cual los autoriza a acceder a un recurso específico, sin utilizar su usuario y contraseña. Una vez que su token fue obtenido, el usuario puede ofrecer su token (que permite el acceso a un recurso específico por un periodo de tiempo) a un sitio remoto. De esta manera, utilizando alguna forma de autenticación: un encabezado, una solicitud GET o POST, o una cookie de

algún tipo, el sitio puede determinar qué nivel de acceso se debe permitir a la solicitud en cuestión.

Esta solución brinda potenciales ventajas, entre ellas:

- Eficiencia y simplicidad de consultas, basta con comparar por un solo campo de la base de datos o aun con menores costos, utilizar un middleware de caché de usuarios, los cuales permiten resolver la consulta sin necesidad de acceder a la base de datos.
- Proveer un mecanismo de autenticación sin estado (stateless), lo que permite una mayor escalabilidad, permitiendo utilizar la misma API para distintos clientes (web, iOS, android, etc.).
- Añade un método mas de seguridad, ya que evita vulnerabilidades del tipo CSRF (Cross-Site Request Forgery) las cuales pueden manipular la sesión que se envía al backend.
- Simple control de acceso a las funciones, basta con identificar el usuario al cual se le asignó el token para poder chequear si el mismo se encuentra autorizado para realizar esa operación. Control sobre los tokens generados en caso de filtración. Esto es, si el token de acceso de un usuario fue interceptado y/o filtrado es posible darlo de baja y generar uno nuevo, a su vez, el token permite un mecanismo de expiración el cual invalida el token después de determinado periodo de tiempo, lo que garantiza que el mismo va a cambiar cada cierto tiempo y no va a permanecer fijo.

En Transcriptor, para obtener un nuevo token de autenticación el usuario debe enviar sus credenciales al backend para que este lo genere. A continuación se incluyen la solicitud y la respuesta HTTP donde se obtiene dicho token:

```
1 POST /api/login
2
3 {
4   "username": "test@transcriptor.edu.ar",
5   "password": "ilovetranscribe"
6 }
```

Figura 7.7: Solicitud HTTP al servicio de inicio de sesión.

```

1  {
2      "status": "OK",
3      "message": "Welcome to Transcriptor!",
4      "data": {
5          "id": 2,
6          "login": "administrator",
7          "display_name": "administrator",
8          "print_name": null,
9          "email": "test@transcriptor.edu.ar",
10         "owner": true,
11         "admin": true,
12         "created_at": "2019-09-28T16:35:52.000Z",
13         "updated_at": "2020-03-30T06:53:12.000Z",
14         "remember_token_expires_at": null,
15         "location": null,
16         "website": null,
17         "about": null,
18         "account_type": null,
19         "paid_date": null,
20         "guest": null,
21         "slug": "administrator",
22         "authentication_token": "Hhyo-pyt-pNWgXPkxAB_",
23         "authentication_token_created_at": "2020-03-30T06:53:12.000Z",
24         "rank": "Visitor"
25     },
26     "alert": null
27 }

```

Figura 7.8: Respuesta HTTP al servicio de inicio de sesión.

En la figura [7.8](#) podemos ver la respuesta a la solicitud de la figura [7.7](#). En ella podemos visualizar que dentro del objeto “data” se encuentra la clave “authentication_token”, cuyo valor representa el token de autenticación a utilizar en las posteriores solicitudes a la API.

Una vez obtenido dicho token, la aplicación cliente (frontend) se encargará de guardarlo y realizar las solicitudes al backend incluyéndolo en el encabezado (header) *Authorization* de la siguiente manera:

Token HhyopytpNWgXPkxAB_

De esta forma la API recibe la solicitud, realizando las validaciones de autenticación correspondientes utilizando el token incluido.

URLs Públicas

Cada controller puede definir los métodos que serán considerados públicos, de manera que no se incluyan en la validación de la sesión, los mismos serán especificados en el método “public_actions”, el cual sobrescribe el método de la superclase *ApiController*.


```
def public_actions
  return [:show, :show_works]
end
```

Figura 7.9: Ejemplo de definición de servicios públicos

Donde cada elemento del arreglo posee un símbolo (Symbol) correspondiente al nombre del método el cual será excluido de la validación. En caso de que no se sobrescriba el método presentado se validaran todas las acciones definidas por el controller.

Métodos de la API

Además de los ya mencionados, la API de transcriptor define varios métodos que permiten realizar las operaciones básicas de CRUD (*Crear, Leer, Actualizar y Borrar*, del original en inglés: *Create, Read, Update and Delete*) de las principales entidades que integran la plataforma, métodos de registro de actividad de usuario y métodos que desencadenan acciones de procesamiento de datos y flujo de información dentro del modelo de negocio. Estas operaciones o *endpoints* de la API de Transcriptor están definidos de la siguiente manera:

Cuadro 7.1: Métodos de la API de Transcriptor

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/login		POST	- username - password	Inicia la sesión utilizando las credenciales recibidas, obtiene los datos del usuario, entre ellos el token de autenticación para realizar los llamados posteriores
/api/registration	-	POST	- login - display_name - email - password	Registra un nuevo usuario a la plataforma utilizando como identificadores de la cuenta al parámetro login y email
		PUT	- login - display_name - email - password - website - about	Modifica el perfil del usuario, permitiendo completar datos como el sitio web o una breve descripción de su rol
/api/users/password		POST	- email	Solicita la recuperación de contraseña del usuario en caso de olvido, al ingresar su email se envía un correo a su casilla con el link de recuperación
	- confirm	POST	-reset_password_token -password -password_confirmation	Recuperación de contraseña, la misma esta asegurada con un token de blanqueo de contraseña, si el mismo no se corresponde la acción es inválida

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/user	badges	GET	-	Obtiene las insignias ganadas por el usuario por las acciones realizadas en la plataforma
	metagame/info	GET	-	Obtiene la información de juego del usuario, esto es el rango y estadísticas de colaboración orientadas a la ludificación
/api/dashboard	guest	GET	-	Lista las actividades recientes de la plataforma para la pantalla inicial pública
	owner	GET	-	Lista las actividades realizadas en las colecciones supervisadas por el usuario
	owner/data	GET	-	Obtiene los datos para mostrar en la pantalla inicial de transcriptor, colecciones del usuario, proyectos y datos estadísticos del sitio
/api/deeds	-	GET	-	Lista las actividades recientes del usuario
	collection/collection_id	GET	-	Lista las actividades recientes de la colección
	work/work_id	GET	-	Lista las actividades recientes del proyecto
/api/upload	/api/upload	POST	- document_upload	Permite la carga de documentos, el parámetro document_upload es un campo de tipo file, el cual puede utilizarse para subir en un formulario multipart/data
/api/collection	-	POST	- title - description - slug (opcional)	Crea una nueva colección con el título y la descripción recibidos
	collection_id	GET	-	Obtiene la información de la colección con id igual a collection_id
		PUT PATCH	- title - description - slug (opcional)	Actualiza la colección con id igual a collection_id con el título y la descripción recibidos
		DELETE	-	Borra la colección indicada en el parámetro collection_id
	list	GET	-	Lista las colecciones existentes
	list_own	GET	-	Lista las colecciones que el usuario creó
	collection_id/works	GET	-	Obtiene la información de los proyectos (works) asociados a la colección

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/work	-	POST	- title (opcional) - description (opcional) - physical_description (opcional) - document_history (opcional) - permission_description (opcional) - location_of_composition (opcional) - author (opcional) - transcription_conventions (opcional) - collection_id - slug (opcional)	Crea un nuevo proyecto con los parámetros recibidos asociándolo a la colección correspondiente a collection_id
	work_id	GET	-	Obtiene la información del proyecto con id igual a work_id
		PUT PATCH	- title (opcional) - description (opcional) - physical_description (opcional) - document_history (opcional) - permission_description (opcional) - location_of_composition (opcional) - author (opcional) - transcription_conventions (opcional) - collection_id - slug (opcional)	Actualiza la colección con id igual a work_id con los datos recibidos
		DELETE	-	Elimina el proyecto indicado en el parámetro work_id
	work_id/pages	GET	-	Obtiene las páginas asociados al proyecto
/api/page	-	POST	- title - work_id	Crea un nuevo proyecto con los parámetros recibidos asociándolo al proyecto correspondiente a work_id
	page_id	GET	-	Obtiene la información de la página con id igual a page_id
		PUT PATCH	- title - work_id	Actualiza la pagina con id igual a page_id con los datos recibidos
		DELETE	-	Elimina la página indicada en el parámetro page_id
	page_id/layers	GET	-	Obtiene las capas de transcripción asociadas a la página

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/page-version	page-version_id	GET	-	Obtiene la información de una versión del texto transcriptor de una página indicando su page-version_id
	page_id/list	GET	-	Lista las versiones del texto transcrito de una página indicada en page_id
/api/layer	-	POST	- name - page_id	Crear una nueva capa de transcripción con nombre de capa especificado en name asociándola a la página correspondiente a page_id
	-layer_id	GET	-	Obtiene la información de la capa con id igual a layer_id
		PUT PATCH	- name - page_id	Modifica la información de la capa de transcripción correspondiente a layer_id
		DELETE	-	Elimina la página indicada en el parámetro layer_id
	layer_id/marks	GET	-	Obtiene las marcas asociadas a la capa
/api/mark	-	POST	- page_id - layer_id - transcription_text (opcional) - translation_text (opcional) - semantic_text (opcional) - schema_type (opcional) - coordinates - shape_type	Crear una nueva marca guardando el arreglo de coordenadas recibidas y genera una nueva contribución basándose en el texto de los campos transcription_text, translation_text, semantic_text, para asociarla a la página y capa correspondientes
	mark_id	GET	-	Obtiene la información de la marca con id igual a mark_id
		PUT PATCH	- transcription_text (opcional) - translation_text (opcional) - semantic_text (opcional) - schema_type (opcional)	Modifica el texto de la contribución asociada a la marca correspondiente a mark_id
		DELETE	-	Elimina la marca indicada en el parámetro mark_id
	mark_id/votes	GET	-	Atajo para poder obtener la cantidad de votos a la contribución de una marca
	mark_id/transcriptions	GET	-	Obtiene las transcripciones asociadas a la capa

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/transcription	-	POST	- text - mark_id	Crea una nueva transcripción utilizando el texto y la asocia a la marca indicada
	transcription_id	GET	-	Obtiene la información de la transcripción correspondiente a transcription_id
		PUT PATCH	- text - mark_id	Modifica el texto de la transcripción correspondiente a transcription_id
		DELETE	-	Elimina la transcripción indicada en el parámetro transcription_id
/api/translation	-	POST	- text - mark_id	Crea una nueva traducción utilizando el texto y la asocia a la marca indicada
	translation_id	GET	-	Obtiene la información de la traducción correspondiente a translation_id
		PUT PATCH	- text	Modifica el texto de la traducción correspondiente a translation_id
		DELETE	-	Elimina la traducción indicada en el parámetro translation_id
/api/contribution	contribution_id/like	GET	-	Genera una acción de voto positivo (like) sobre la contribución indicada en contribution_id
	contribution_id/dislike	GET	-	Genera una acción de voto negativo (dislike) sobre la contribución indicada en contribution_id
	contribution_id/vote	GET	-	Obtiene la información del estado del voto (positivo, negativo, neutro) sobre la contribución indicada en contribution_id
/api/transcribe	page_id/transcribe	POST	- page - page.source_text - needs_review - mark_blank	Permite guardar el texto transcrito utilizando referencias a las marcas creadas, aceptando tags HTML que permiten darle formato, además permite guardar opciones de estado de la transcripción como "necesita revisión." "pagina en blanco"
	page_id/translate	POST	- page - page.source_translation - needs_review - mark_blank	Permite guardar el texto traducido utilizando referencias a las marcas creadas, aceptando tags HTML que permiten darle formato, además permite guardar opciones de estado de la traducción como "necesita revisión." "pagina en blanco"

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/forum	classname/element_id	GET	-	Obtiene la información del foro utilizando el element_id al cual hace referencia y su clase, si el mismo no existe lo da de alta
	forum_id	GET	-	Obtiene la información del foro correspondiente a forum_id
	forum_id/publications	GET	-	Obtiene las publicaciones asociadas al foro
/api/publication	-	POST	- text - forum_id	Crea una nueva publicación con el texto recibido en el parámetro text asociándola al foro correspondiente a forum_id
	publication_id	GET	-	Obtiene la información de la publicación en un foro correspondiente a publication_id
		PUT PATCH	- text - forum_id	Obtiene la información de la publicación en un foro correspondiente a publication_id
	parent_publication_id/list	GET	-	Obtiene las respuestas (publicaciones hijas) asociadas a una publicación
	publication_id/like	GET	-	Genera una acción de voto positivo (like) sobre la publicación indicada en publication_id
	publication_id/dislike		-	Genera una acción de voto negativo (dislike) sobre la publicación indicada en publication_id
	publication_id/vote	GET	-	Obtiene la información del estado del voto (positivo, negativo, neutro) sobre la publicación indicada en publication_id

7.4.5. Persistencia

Transcriptor utiliza como motor de base datos relacional a MySQL debido a su flexibilidad y escalabilidad, su amplia disposición en sistemas operativos y su facilidad de instalación y uso desde Ruby a través de la gema mysql2.

Por otro lado Ruby On Rails provee el ORM Active Record, el cual se encarga del mapeo del modelo relacional a objetos y permite realizar las operaciones de lectura y escritura necesarios para proveer una correcta interfaz con la base de datos. Active Record esta basado en el patrón de diseño arquitectónico que lleva el mismo nombre descripto por Martin Fowler como "Un objeto que envuelve una tupla de una tabla o vista en una base de datos, encapsulando el acceso a la misma y agregando lógica de negocio a esos datos.", de esta manera Active Record facilita la creación y el uso de los objetos del dominio en los cuales los datos que representen requieran de almacenamiento persistente en la base de datos.

A continuación se presentará la estructura y la relación de las tablas mas importantes de la base de datos del Transcriptor.

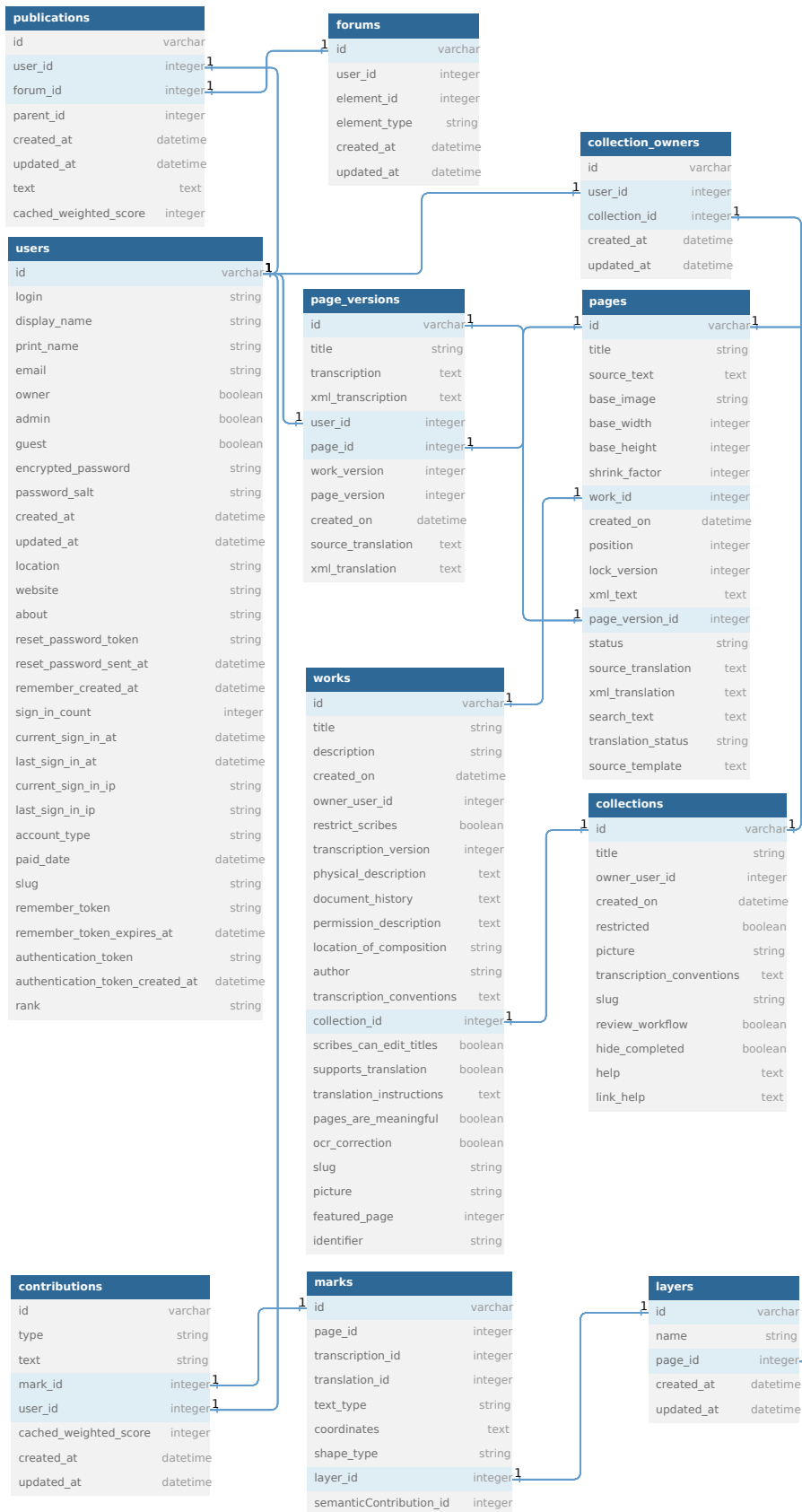


Figura 7.10: Diagrama Entidad-Relación de la base de datos de Transcriptor

7.4.6. Gamificación

En este apartado nos vamos a centrar en la implementación de las estrategias de gamificación en Transcriptor, las cuales fueron introducidas en la sección [6.2.7](#). Estas estrategias se desarrollaron haciendo uso del framework Metagame, el cual posee una API Rest que permite registrar distintos tipos de actividades que se realizan en la plataforma y se distinguen en «tipos de transcriptor». A medida que el usuario va realizando aportes sobre Transcriptor este va acumulando experiencia, la cual se ve reflejada en medallas y promoción de rangos. La obtención de estas medallas a su vez se traducen en un ascenso a través de la jerarquía de rangos.

Integrando metagame-api

Para comenzar a registrar actividades realizadas por los usuarios necesitamos, en primer lugar, contar con un conector que provea una interfaz con el servicio Metagame. Para poder interactuar con el servicio, el framework brinda una API Rest, a continuación presentamos los principales métodos:

Ruta	Método	Parámetros	Descripción
/activities	POST	- email (obligatorio) - project (obligatorio) - event (obligatorio) - count (obligatorio)	Registra una actividad de usuario con el tipo especificado en event (contribution, special, dissemination, reinforcement) y la cantidad de veces especificada en count
/player_info	GET	- email (obligatorio)	Obtiene la información del jugador: insignias, rango, y perfil
/player	POST	- email (obligatorio)	Crea un nuevo jugador de Metagame. El parámetro email es obligatorio, dado que es un identificador
	DELETE PUT PATCH	- email (obligatorio) - email (obligatorio)	Elimina a un jugador Actualiza la información de un jugador

Para realizar las solicitudes a estos servicios el conector tendrá que implementar un cliente HTTP, el cual encapsulará la lógica que involucra el llamado a cada uno de estos métodos. De esta manera se generó la clase MetagameClient que abstrae la complejidad de la integración y permite la invocación a los métodos de la metagame-api de manera transparente enviando los parámetros necesarios para el registro y procesamiento de las actividades realizadas por el usuario. A su vez el MetagameClient hereda comportamiento de la clase HTTPClient (ver figura [7.11](#)) la cual es un cliente HTTP genérico que se encarga de realizar cualquier tipo de solicitud y a su vez procesar la respuesta obtenida, haciendo uso del cliente Net::HTTP nativo en el lenguaje Ruby.



Figura 7.11: Diagrama de clases del cliente de gamificación.

A modo de ejemplo y para mayor comprensión de la descripción de la integración se incluirán algunos fragmentos del código del conector:

```

1  def do_post(path, params = {}, bodyParams = {})
2  send_request(Net::HTTP::Post, path, params, bodyParams)
3  end
4
5  def do_get(path, params = {})
6  send_request(Net::HTTP::Get, path, params)
7  end
8
9  def send_request(methodClass, path, params = {}, bodyParams = {})
10 uri=URI(@baseUrl)
11 uri.path=path
12 uri.query=URI.encode_www_form(params)
13
14 req = methodClass.new(uri)
15 # Set headers in request
16 @headers.each do |key, value|
17   req[key] = value
18 end
19 req.set_form_data(bodyParams)
20
21 res = Net::HTTP.start(uri.hostname,uri.port, :use_ssl => uri.scheme == 'https') {|http|
22   http.request(req)
23 }
24 process_response(res)
25 end
26
27 def process_response(response)
28 case response
29 when Net::HTTPSuccess
30   JSON.parse(response.body, object_class: OpenStruct)
31 when Net::HTTPUnauthorized
32   {'error' => "#{response.message}: username and password set and correct?"}
33 when Net::HTTPServerError
34   {'error' => "#{response.message}: try again later?"}
35 else
36   {'error' => response.message}
37 end
38 end

```

Figura 7.12: Implementación del envío de solicitudes a la API de metagame.

En la figura [7.12](#) podemos ver la implementación del método *do_post* de la clase HTTP-Client el cual se encarga de realizar una request HTTP utilizando el método POST, en él se deriva toda la lógica de envío de la solicitud al método *send_request*, donde se realiza finalmente la solicitud HTTP utilizando la librería *Net::HTTP*. Este principio de delegación permite implementar el llamado al resto de los métodos HTTP, como es el caso del método GET en la función *do_get*.

```

1  def make_activity(params)
2  |    do_get("/activities",params)
3  end

```

Figura 7.13: Implementación del método `make_activity` de la clase `MetagameClient`.

Siguiendo con la implementación de la integración, se presenta el método `make_activity` de la clase `MetagameClient` que registra la actividad del usuario en el servicio de gamificación [7.13](#). Esto es posible ya que la clase `MetagameClient` aprovecha la herencia de los métodos de la superclase `HTTPClient` haciendo uso del método `do_get` para realizar una solicitud al endpoint `/activities` de Metagame.

Finalmente Transcriptor cuenta con una ultima clase llamada `GamificationHelper` la cual se encarga de realizar la interfaz entre el `MetagameClient` y la lógica de negocio de Transcriptor, encargándose de notificarle si el usuario recibe una nueva medalla por la acción que acaba de realizar. A continuación un ejemplo de uno de sus métodos

```

1  def self.registerEvent(email)
2  |    begin
3  |      activity={"email":email,"project":@@project,"event":"login","count":1}
4  |      @@metagameClient.make_activity(activity)
5  |      return self._badge_notification("i-was-here")
6  |    rescue Exception => e
7  |      return nil
8  |    end
9  end

```

Figura 7.14: Implementación del método `registerEvent` de la clase `GamificationHelper`.

En la figura [7.14](#) podemos ver la implementación del método estático `registerEvent` de la clase `GamificationHelper`, que describe la funcionalidad de notificar al `MetagameClient` que un usuario se acaba de registrar y además obtuvo la medalla “i-was-here” debido a ser la primera vez que ingresa a la plataforma. Este método es llamado en el servicio de registración de Transcriptor.

El flujo descrito anteriormente se cumple para el resto de las actividades realizadas en transcriptor, utilizando el `GamificationHelper` como nexo entre Transcriptor y Metagame. Esta clase es también utilizada para obtener y listar las medallas obtenidas por el usuario, haciendo la correspondiente consulta a la `metagame-api`.

7.4.7. Transcripción semántica

Este apartado contiene conceptos y definiciones trabajados en la sección Capas de transcripción semántica del capítulo [6.2.4](#) de esta tesis, por lo que se le sugiere al lector tenga presente los mismos para facilitar la comprensión de los temas tratados a continuación.

Como se describió anteriormente, Transcriptor permite describir los elementos presentes en un manuscrito utilizando sintaxis semántica estructurada, esto se logra a través de

la implementación de un nuevo tipo de contribución a la que se denominó Contribución Semántica (*SemanticContribution*), las cuales se encargan de representar y mantener una referencia a esta información. Estas contribuciones semánticas están compuestas por dos elementos principales:

- Anotación: también llamadas marcas (*Mark*), son las que mantienen la relación posicional en el manuscrito, utilizadas también para el flujo de transcripción convencional.
- Entidad semántica: estructuras que contienen la información semántica asociada a la anotación, se corresponden con un tipo definido por la ontología utilizada y por consiguiente incorpora valores a las propiedades de dicho tipo (p.e. para una entidad de tipo *Persona*, a la propiedad *nombre* define como valor *Juan*).

Las ontologías proveen un conjunto de conceptos y las relaciones entre ellos que se utilizan para describir un dominio específico del conocimiento, esta información se emplea por el cliente (frontend), para construir las entidades semánticas que luego serán almacenadas en el core (backend) de Transcriptor. En base a este flujo se generaron un conjunto de servicios REST, los cuales permiten la creación de anotaciones semánticas utilizando múltiples ontologías precargadas (ver figura 7.9).

Métodos de la API Semántica de transcriptor

En adición a los métodos de la API REST presentados en el apartado 7.4.4, Transcriptor incorpora un set de *endpoints* para el manejo de colecciones y anotaciones semánticas, lo que funciona como soporte para el flujo de transcripción semántica.

Cuadro 7.9: Métodos de la API Semántica de Transcriptor

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/ontology	-	GET	-	Lista las ontologías configuradas en la plataforma
		POST	- name - description - url - domainkey - rangekey - prefix - literal_type - class_type - graph_file -ontology_datatypes_attributes	Guarda una nueva ontología configurada con los parámetros y el archivo de definición especificados
	ontology_id	PUT	- name - description - url - domainkey - rangekey - prefix - literal_type - class_type - graph_file -ontology_datatypes_attributes	Modifica la configuración de la ontología indicada en el parámetro <i>ontology_id</i>
		DELETE	-	Elimina la configuración de la ontología indicada en el parámetro <i>ontology_id</i>

Ruta	Parámetros de Ruta	Método	Parámetros Cuerpo	Descripción
/api/ontology	list_classes	GET POST	- ontology_id - parent	Lista jerarquía de clases de la ontología <i>ontology_id</i> tomando como base la clase especificada en el parámetro <i>parent</i>
	list_properties	GET POST	- ontology_id - parent	Lista las propiedades admitidas para la clase enviada en el parámetro <i>textitclass</i>
	list_relations	GET POST	- ontology_id - parent	Lista las relaciones admitidas para la clase enviada en el parámetro <i>textitclass</i>
	search_loaded_components	GET POST	- search_text - semantic_component	Realiza una búsqueda en la base semántica y lista todos los elementos cargados que coincidan el texto <i>textitsearch_text</i> filtrando coincidencias de tipo "clase" (<i>class</i>) y relaciones (<i>relation</i>) usando el valor del parámetro <i>semantic_component</i>
/api/semantic_entity	add_relation	POST	- subject_id - predicate_id - object_id	Relaciona las entidades con id igual a <i>subject_id</i> y <i>object_id</i> utilizando la propiedad correspondiente al id <i>predicate_id</i>
	describe	GET	- entity_id	Obtiene información de la entidad semántica con identificador igual a <i>entity_id</i> incluyendo su propiedades y principales relaciones.
	list	POST	- filter - labelValue (opcional) - propertyValue (opcional) - propertyName (opcional) - entityTypeLike (opcional)	Lista las entidades semánticas guardadas filtrando según los criterios especificados en el objeto recibido en el parámetro <i>filter</i>
	list_marks	POST	- filter - entity_id (opcional) - labelValue (opcional) - propertyValue (opcional) - propertyName (opcional) - entityTypeLike (opcional)	Retorna un listado de marcas (contribuciones) que hagan referencia a una entidad semántica que cumpla con las condiciones especificadas en el parámetro <i>filter</i>

Persistencia de Datos Semánticos

De la misma manera en que se realiza la persistencia de las entidades del modelo, la información descripta en formato semántico también conforma una estructura persistente, la cual propone un nuevo flujo y metodología para lograr este objetivo. En este contexto, para almacenar dichos datos es necesario de un servicio especializado el cual pueda interpretar y generar las relaciones dinámicas creadas a través de la transcripción semántica de los elementos del manuscrito.

Para implementar la persistencia de estos datos se utilizó Virtuoso Universal Server. Este producto, permite almacenar los triplets (véase capítulo [6.2.4](#)) que conforman las definiciones de los datos Semánticos que el usuario carga, conformando un grafo de relaciones entre estas entidades, el cual permite ser consultado en cualquier momento a

través de servicios de obtención de datos. Virtuoso provee una API REST que se utiliza para realizar todas las operaciones sobre los datos almacenados, estos servicios a su vez se valen de un lenguaje de consulta especializado en consultas de grafos RDF, que permite aplicar condiciones y proyectar la información tal como lo hace el lenguaje de consultas SQL. Transcriptor encapsula la lógica de integración a Virtuoso en una clase Helper, la cual posee los métodos necesarios para comunicarse con la API REST de este servicio y de esta manera realizar las operaciones de alta y consultas sobre la información almacenada.



Figura 7.15: Diagrama de clases de la integración a virtuoso

Como se puede observar en el la figura [7.15](#) la clase `SemanticHelper` actúa como interfaz a través de la cual la plataforma realiza las peticiones al cliente concreto, proporcionándole abstracción al modelo. Finalmente, el conector a Virtuoso (clase `VirtuosoClient`)

hace uso de de la clase HTTPClient la cual brinda las funciones para realizar las solicitudes HTTP que hacen posible la integración.

Procesamiento de Ontologías

Para acceder a los tipos definidos por cada ontología, se definieron tres servicios REST, según el tipo de elemento por el que esta compuesto:

Obtención de clases (list_classes): Obtiene un listado de las clases presentes dentro de una ontología precargada. La clase helper se encarga de realizar la consulta SPARQL a Virtuoso y obtiene una proyección de todos los nombres de cierto nivel, determinado por el parámetro *parent* enviado, incluye además en la respuesta un listado de sus ancestros, lo cual permite la navegación del listado de clases de manera ordenada, avanzando a través de los hijos de cada clase (figura 7.16).

```
1 def list_classes(ontology_id, parent = nil, include_parent = false)
2   begin
3     # Obtiene la ontología precargada en base a los parametros
4     ontology = getOntology(ontology_id, parent)
5     results = []
6     # Genera una nueva instancia del cliente SPARQL
7     # para realizar la consulta con los datos de la ontología obtenida
8     sparql = SPARQL::Client.new("#{@host}/sparql", { graph: ontology.url })
9     # Analiza los filtros de antecesores en la jerarquía dependiendo del parametro parent y include_parent
10    parentInclusionFilter = include_parent ? '*' : ''
11    if parent
12      parentFilter = parent ? "?classId rdfs:subClassOf#{parentInclusionFilter} #{formatId(parent)} ."
13    else
14      parentFilter = "FILTER NOT EXISTS { ?classId rdfs:subClassOf ?parentClass . }"
15    end
16    # Construye la nueva consulta utilizando los filtros y los datos de la ontología
17    statement = "
18    select ?label ?comment ?classId
19    where {
20      ?classId rdfs:label ?label.
21      optional { ?classId rdfs:comment ?commentOptional }.
22      ?classId rdf:type #{ontology.class_type}.
23      bind(coalesce(?commentOptional, '') As ?comment) .
24      FILTER NOT EXISTS {
25        | ?classId rdf:type #{ontology.literal_type}
26      }.
27      #{parentFilter}
28    }"
29    # Ejecuta la consulta utilizando el cliente SPARQL
30    query = sparql.query(statement)
31    # Mapea los resultados a un Map y los agrega a la colección de resultados para retornarlos.<<
32    query&.each_solution do |solution|
33      results.push({
34        | id: solution[:classId].value, label: solution[:label].value, comment: solution[:comment].value})
35    end
36    return results
37  rescue => exception
38    return []
39  end
40 end
```

Figura 7.16: Implementación del servicio list_classes

Obtención de propiedades simples (list_properties): Dado el nombre de una clase de la ontología enviado por parámetro retorna un listado de sus propiedades que soportan tipos simples: numéricos, cadena de caracteres y valores booleanos.

Obtención de propiedades complejas/relaciones (list_relations): De la misma manera que el servicio anterior, utilizando el nombre de la clase consulta por sus propiedades, pero en este caso teniendo en cuenta las complejas. Estas son las cuales reciben como valor tipos complejos, en general otras instancias de las clases de la ontología, estas pueden ser Person, Book, Manuscript, etc.

Estos servicios intervienen para que el cliente pueda obtener todos los datos para construir las descripciones semánticas de los elementos del documento. Una vez realizada esa descripción, la siguiente tarea que debe realizar Transcriptor-Backend será manejar la persistencia de las mismas, donde intervendrán otras herramientas presentadas en la próxima sección.

Persistencia aplicada al flujo de transcripción

Como ejemplo del uso de este conector se tomará el flujo de creación de una contribución semántica:

```
1 class SemanticContribution < Contribution
2   after_create :insert_semantic_register
3
4   def insert_semantic_register()
5     SemanticHelper.insert(self.text)
6   end
7 end
```

Figura 7.17: Definición de la clase SemanticContribution

Active Record provee de métodos Callback los cuales se ejecutan cuando se ocurren determinados eventos en el ciclo de vida de una entidad. En el caso de las contribuciones semánticas, se utilizó del método `after_create`, el cual se invoca luego de que una entidad es persistida, logrando que, una vez que el controlador procese la solicitud HTTP e inicie el proceso de guardado de una nueva contribución, se ejecute el método `insert` de la clase `SemanticHelper`.

```
1 def insert(jsonld_string)
2   rdf_string = jsonldToRdf(jsonld_string)
3   query = "INSERT IN GRAPH <#{ @graph }> { #{rdf_string} }"
4   @httpClient.do_post(@collection, {}, query)
5 end
```

Figura 7.18: Método insert de la clase VirtuosoClient

El método `insert` es el encargado de generar y persistir una nueva entidad semántica en Virtuoso. Esta función recibe como parámetro los datos semánticos estructurados en formato JSONLD, el cual facilita la lectura y su confección en las aplicaciones clientes, pero

no es el soportado por Virtuoso de manera nativa y en consecuencia es necesario realizar conversión de los datos a formato RDF, véase figura 7.18 línea 2. Una vez obtenidos los datos en formato RDF, se construye una consulta SPARQL utilizando estos valores y se envían al servicio a través del método `do_post` del `HTTPClient` (línea 4), que simplemente realiza un post con los datos proporcionados a la URL especificada por parámetro. De esta forma el proceso de creación de contribuciones se unifica, generando los correspondientes registros, tanto en la base de datos de transcriptor como en el grafo de datos de Virtuoso.

7.4.8. Internacionalización en Servidor (Server-Side i18n)

Se define internacionalización al proceso en el cual se diseña a un software para adaptarse a diferentes regiones y lenguajes sin necesidad de que este necesite de un cambio de código. Este concepto se encuentra íntimamente ligado con la Localización, que se describe como la adaptación del software a una región específica empleando componentes que permiten definir un esquema de traducción de textos y conversión de monedas denominados Locales. Transcriptor-Backend implementa esta lógica utilizando la API de internacionalización de Ruby On Rails, que permite definir archivos de traducciones (también llamados de localización) para cada idioma o región, constituidos por una estructura de clave-valor (ver figura 7.19) donde la clave representa nombre descriptivo del valor textual a guardar.

```
1 # es.yml
2
3 es:
4   api:
5     default:
6       ok: "La operación se realizó exitosamente"
7       error: "Error del servidor, inténtelo más tarde."
8     session:
9       not_allowed: "Para acceder a este contenido debes iniciar sesión"
10    login:
11      success: "¡Bienvenido a Transcriptor!"
12      fail: "Usuario o contraseña no válidos, vuelva a intentarlo."
13    reset_password:
14      request:
15        success: "Recibirás un email con instrucciones para cambiar tu contraseña en unos minutos."
16        error: "El email ingresado no corresponde con ningún usuario registrado"
17      confirm:
18        success: "Tu contraseña fue modificada correctamente. Has iniciado sesión."
```

```
1 # en.yml
2
3 en:
4   api:
5     default:
6       ok: "The operation was successfully completed"
7       error: "server error, contact the administrator"
8     session:
9       not_allowed: "To access this content you must log in to this site"
10    login:
11      success: "Welcome to Transcriptor!"
12      fail: "Username or password not valid, try again."
13    reset_password:
14      request:
15        success: "You will receive an email with instructions on how to reset your password in a few minutes."
16        error: "The email entered does not correspond to any registered user"
17      confirm:
18        success: "Your password was changed successfully. You are now signed in."
```

Figura 7.19: Fragmentos de archivos de traducción, español (es.yml) e ingles (en.yml)

Del lado del código, para obtener el texto correspondiente al idioma solicitado se refe-

rencia al método estático `translate` de la clase `I18n` enviando como parámetro la clave de del texto a referenciar, esta se busca en el archivo correspondiente al idioma y obtiene el valor al cual identifica la clave. A continuación se presenta un ejemplo de lo tratado:

```
1  class ResponseWS
2    attr_accessor :status, :message, :data
3
4    def initialize(status,message,data,alert)
5      @status = status
6      @message = I18n.t(message, :default => message)
7      @data = data
8      @alert = alert
9    end
10
11   def self.ok(message,data,alert = nil)
12     ResponseWS.new("OK",message,data,alert)
13   end
14
15   def self.default_ok(data,alert = nil)
16     ResponseWS.ok('api.default.ok',data,alert)
17   end
18
19   def self.error(message,data,alert = nil)
20     ResponseWS.new("ERROR",message,data,alert)
21   end
22
23   def self.default_error(alert = nil)
24     ResponseWS.error('api.default.error',nil,alert)
25   end
26 end
```

Figura 7.20: Fragmento de la clase `ResponseWrapper` se traduce la clave al texto de respuesta para el usuario.

En la figura [7.20](#) se muestra un fragmento de la clase `ResponseWrapper` donde se puede observar, en la línea 6, como se hace referencia en su inicialización a una clave recibida para poder obtener el valor textual en un idioma determinado, que representa el mensaje a mostrar al usuario en la respuesta de la API. Esta técnica logra una independencia entre la lógica de negocio y los valores de los textos a mostrar, permitiendo mayor control y mantenimiento (todo el contenido textual esta definido en un solo archivo por cada idioma) y además facilita una dinámica de cambio de lenguaje *on the fly* sin necesidad de recompilar o generar cambios en la estructura del proyecto.

Detectando el idioma del usuario

Para lograr la traducción de las claves se necesita de un idioma al cual referenciaremos para realizar la búsqueda en el archivo de traducciones, para seleccionarlo Transcriptor-API cuenta con dos alternativas:

- **Parámetro locale:** todos los endpoints de la API están habilitados para recibir el parámetro locale en su cadena de consulta (query string) especificando el lenguaje a utilizar para la respuesta:

/api/login?locale=en	/api/login?locale=es
<pre>1 { 2 "status": "OK", 3 "message": "Welcome to Transcriptor!", 4 "data": { 5 "id": 2, 6 "login": "administrator", 7 "display_name": "administrator", 8 "print_name": null, 9 "email": "transcriptor_boss@hotmail.com",</pre>	<pre>1 { 2 "status": "OK", 3 "message": "¡Bienvenido a Transcriptor!", 4 "data": { 5 "id": 2, 6 "login": "administrator", 7 "display_name": "administrator", 8 "print_name": null, 9 "email": "transcriptor_boss@hotmail.com",</pre>

Figura 7.21: Ejemplo de utilización de parámetro locale y cambio dinámico del idioma del mensaje mostrado al usuario.

- **Encabezado Accepted-Language:** Por defecto, todos los navegadores envían en cada solicitud el encabezado HTTP Accepted-Language que especifica el idioma en el cual se encuentra configurado en base a este valor se selecciona el idioma a utilizar.

"Accept-Language": en	"Accept-Language": es
<pre>1 { 2 "status": "OK", 3 "message": "Welcome to Transcriptor!", 4 "data": { 5 "id": 2, 6 "login": "administrator", 7 "display_name": "administrator", 8 "print_name": null, 9 "email": "transcriptor_boss@hotmail.com",</pre>	<pre>1 { 2 "status": "OK", 3 "message": "¡Bienvenido a Transcriptor!", 4 "data": { 5 "id": 2, 6 "login": "administrator", 7 "display_name": "administrator", 8 "print_name": null, 9 "email": "transcriptor_boss@hotmail.com",</pre>

Figura 7.22: Ejemplo de utilización de encabezado Accepted-language y cambio dinámico del idioma del mensaje mostrado al usuario.

Estos mecanismo de selección de idioma se encuentran implementados a su vez en el la clase APIController, ya presentada anteriormente, el cual entre otras cosas se encarga de discernir y configurar el lenguaje con el que se va a trabajar para responder a la solicitud.

```

1  def set_locale
2    if params[:locale]
3      I18n.locale = params[:locale]
4    else
5      I18n.locale = extract_locale_from_accept_language_header || I18n.default_locale
6    end
7  end

```

Figura 7.23: Método `set_locale`, encargado de la lógica de selección de idioma.

En la figura [7.23](#) podemos ver la implementación de la función `set_locale` que se encarga de configurar la clase `I18n` para responder utilizando el idioma especificado tanto en el parámetro `locale` como en el encabezado `Accepted-Language`.

7.4.9. Despliegue e Instalación

Tanto para utilizarlo en ambientes de desarrollo o productivos, es necesario instalar y configurar el proyecto de `Transcriptor-Backend` para cada fin. En este apartado vamos a realizar una descripción de los pasos de de instalación y el método de despliegue.

Instalando dependencias

`Transcriptor-Backend` hace uso de muchas herramientas y librerías para realizar sus funciones, desde el núcleo del framework MVC de la aplicación (`RubyOnRails`) hasta librerías de procesamiento de imágenes, lógica de votaciones, parametrización de configuraciones, etc. Para mantener un control de estas dependencias, `Transcriptor` utiliza el manejador de dependencias `Bundler`, el cual permite llevar un registro de instalación de las Gemas (librerías en Ruby) del proyecto para mantener consistencia a los requerimientos de cada ambiente, ya sea desarrollo, prueba o producción. Esto se logra definiendo, al momento del desarrollo, todas las gemas a utilizar en un archivo llamado `Gemfile`. Finalmente, para realizar la instalación de las gemas solo es cuestión de ejecutar el comando `bundle install`, que desencadena la descarga y referenciación de todos los paquetes declarados en el archivo `Gemfile`.

Configurando conexiones

El proyecto posee múltiples archivos de configuración donde se definen los valores de los parámetros necesarios para la integración con otras herramientas y servicios. La configuración inicial está contenida en el archivo `application.yml` que define los valores críticos para el funcionamiento del sistema, como lo son los datos de conexión a la base de datos y al sistema de metajuego (`metagame`). A su vez estos datos pueden definirse en variables de entorno en lugar de un archivo y de esta manera delegar la configuración a la instancia donde se ejecuta la aplicación.

```
1 # Database config
2 DATABASE_HOST: "localhost"
3 DATABASE_USERNAME: "transcriptor"
4 DATABASE_PASSWORD: "transcriptor_db_password"
5
6 #Configuration of metagame
7 METAGAME_PROJECT_NAME: "transcriptor"
8 METAGAME_URL: "https://metagame-api.herokuapp.com"
9 METAGAME_TOKEN: "14a3685d8a4443eb8a6136a38e76a7e7"
```

Figura 7.24: Valores de configuración en aplicacion.yml

En un segundo nivel, se encuentran los archivos de configuración de ambiente, Transcriptor define tres ambientes: Desarrollo (Development), Prueba (Test) y Producción (Production). En cada uno de estos archivos podemos encontrar configuraciones de incumbencia para cada uno de estos ambientes, tales como configuraciones de salida de mails, manejo de cache, nivel de log, URL del host, etc.

Carga de Base de Datos

Para generar las tablas y los datos por defecto, Transcriptor utiliza el esquema de Migraciones (Migrations) de ActiveRecord. Las migrations son archivos de cambios incrementales reversibles, los cuales son utilizados en la base de datos para aplicar modificaciones de una manera estructurada y organizada. De esta manera facilita el desarrollo en conjunto, automatizando la tarea de aplicar los cambios en las bases locales de cada desarrollador del equipo y manteniendo un registro de cambios realizados en la/s bases productivas cada vez que se publique una nueva versión del proyecto. Para poder cargar la estructura de la base de datos de Transcriptor, es necesario ejecutar todas las migraciones existentes en el proyecto, esto se realiza a través del comando `rake db:migrate`. Una vez finalizado el proceso, se habrán generado todas las tablas del modelo en su estado inicial y listas para usar.

Publicar en Servidor

Para poder recibir las solicitudes y procesar las respuestas es necesario desplegar la aplicación en un servidor, el tipo de servidor a utilizar depende del ambiente sobre el cual vamos a utilizar Transcriptor-API. En el caso de utilizarlo en un ambiente de desarrollo, el framework Ruby on Rails incluye un servidor por defecto para este propósito, llamado WEBrick. Para ejecutarlo se utiliza el comando `rails serve`, que iniciara la aplicación en el puerto 3000 del equipo local (localhost) de esta manera podemos tener nuestra aplicación publicada y lista para su uso y desarrollo con poca configuración y de manera ágil. Para utilizarlo en ambientes productivos Transcriptor-API hace uso de un módulo para el servidor web Apache llamado Phusion Passenger (Passenger) el cual provee de un contenedor que se encarga de interpretar y ejecutar el código Ruby haciendo posible la integración con el servidor web y permitir realizar su configuración a través de los archivos de Apache,

centralizando los parámetros de cada herramienta y tratándolas como una sola. Esta infraestructura permite que el encargado de la solicitudes y respuestas HTTP sea Apache, que provee un soporte y performance robusta en estas tareas, y la interpretación y ejecución del código a Passenger.

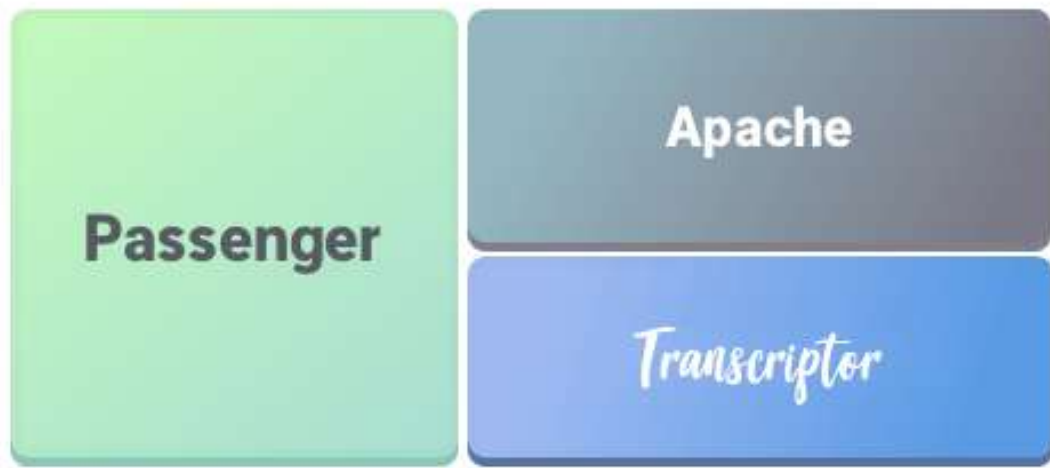
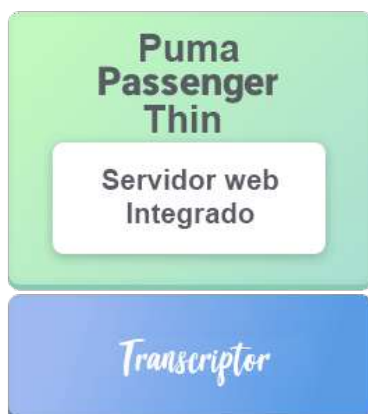


Figura 7.25: Infraestructura de servidor productivo de Transcriptor.

Finalmente cabe mencionar que existen otras alternativas de configuración para ambientes de producción utilizando servidores como Puma o Thin que proveen un funcionamiento similar a Passenger con distintas capacidades de procesamiento y facilidad de integración. A su vez es posible cambiar este esquema de infraestructura utilizando otros servidores web en lugar de Apache como Nginx o los servidores standalone incluidos en Passenger, Puma o Thin.

Integración Standalone



Integración Nginx

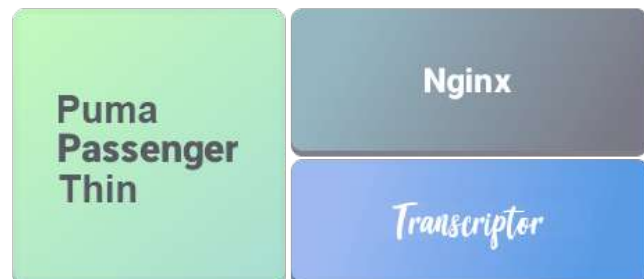


Figura 7.26: Métodos alternativos de configuración del servidor productivo.

7.5. Frontend

La aplicación web o Frontend (a partir de ahora Transcriptor-Frontend) es la capa permite al usuario interactuar con los servicios de Transcriptor-Backend, abstrayéndolo de la lógica que esto conlleva. Utilizando una interfaz web intuitiva, facilita la carga y presentación de los datos que serán enviados a través de solicitudes a los endpoints definidos, creando flujos de pantallas para determinados propósitos: transcribir, listar colecciones, ver las paginas de un proyecto, entre otras.

7.5.1. Tecnologías Utilizadas

- **Angular:** framework para el desarrollo de aplicaciones web basado en el patrón MVC, facilitando la creación y el mantenimiento de aplicaciones web de una sola pagina, esta desarrollado en TypeScript y actualmente es mantenido por Google.
- **Materialize:** framework CSS responsive basado en el concepto de diseño Material Design, provee componentes visuales y herramientas para el maquetado de las vistas del proyecto.
- **Ngx-Materialize:** librería que actúa de envoltorio sobre Materialize implementando componentes Angular que facilitan el uso de materialize en el proyecto, simplificando la integración y brindando servicios de inyección de módulos, inicialización automática y formularios reactivos.
- **Leaflet:** librería Javascript utilizada para construir aplicaciones web de mapas, posee soporte para plataformas web y móvil, haciendo uso de HTML y CSS3.
- **Leaflet Draw:** agrega soporte para renderizar polígonos, vectores y marcadores en los mapas de Leaflet. Ngx-editor: Editor WYSIWYG (What You See Is What You Get, en español, "lo que ves es lo que obtienes") para aplicaciones Angular.
- **RxJS:** librería de programación que hace uso de objetos observables, para facilitar la implementación de llamados asíncronos y/o funciones callback.
- **Ngx-Translate:** librería que provee un conjunto de funciones y un marco de definición y recuperación de archivos de traducción para realizar la internacionalización de componentes Angular.

7.5.2. Tipos de módulos

Transcriptor-Frontend esta implementado utilizando el framework Angular, el que permite generar aplicaciones de una sola pagina (single page) los cuales permiten generar dinamismo y a su vez mayor performance de carga de sus recursos. La arquitectura que plantean este tipo de frameworks es MVC de manera que sus componentes interactúan entre si para generar la presentación, procesamiento y consulta de los datos. En la arquitectura de Transcriptor-Frontend podemos diferenciar 4 componentes principales las cuales es importante conocer su función para luego explicar como actúan en las distintas pantallas de la aplicación:

- **Componente (component):** integran la unidad fundamental y mas básica dentro de la aplicación y representan los bloques con cuales construirla. Cada pantalla de la plataforma esta definida como un componente que a su vez están integrados por tres elementos:

- *Vista (view)*: es el módulo encargado de la visualización del componente define estructura y estilo de la presentación del mismo Controlador (controller): Contiene la lógica interna de funcionamiento del componente y es el encargado de la interacción con otros miembros de la aplicación web.
 - *Otros componentes*: A su vez un componente puede contener a otros componentes los cuales pueden dividirse el trabajo de visualización y procesamiento del componente, delegando la lógica entre estos y facilitando su mantenimiento y reutilización.
- **Servicio (service)**: Módulos encargados de proveer “datos” que la aplicación necesita, se encuentran separados de los componentes para facilitar su mantenimiento y reutilización. Permiten ser inyectados en los controladores de los componentes para que estos les deleguen tareas como: obtención de datos del servidor consumiendo de la API, validación de datos o registrar eventos de cierta índole. Lo que incurre en que el servicio adopte una noción de “capa de modelo”
 - **Modelo (model)**: Objetos asociados a una clase, representan un dato obtenido de la respuesta de Transcriptor-API. Permiten además, definir comportamiento y resolver la lógica de negocio de la misma manera en que lo hace Transcriptor-Backend con sus entidades.
 - **Enrutador (router)**: Permite la navegación desde una vista a la siguiente a medida que el usuario utiliza las distintas tareas de la aplicación, relacionando los distintos componentes con URI’s dentro de la plataforma. De esta manera, ya sea que el usuario haya ingresado una dirección específica o que haya seguido un enlace, le será posible acceder a la pantalla deseada cargando sus recursos bajo demanda.

7.5.3. Cliente HTTP

Como parte fundamental de toda implementación de una aplicación cliente, es importante desarrollar una manera simple y mantenible de comunicarse con sus servicios para garantizar el correcto consumo de la información y, de esta manera poder utilizarla para los fines esperados.

En el caso de Transcriptor, los módulos encargados de realizar la invocación al servicio desde Transcriptor-Frontend son, literalmente, los módulos de servicio o Services. Los Services engloban los distintos llamados a los métodos (endpoints) de la API según la entidad a la que hacen referencia, es así como por ejemplo, los servicios de administración de marcas están agrupados en el Service de Marcas (Mark-Service) o los servicios de colecciones al Service de Colecciones (Collection-Service), etcétera.

Todos los Services que se comunican con Transcriptor-API, a su vez, delegan la lógica de la realización de la solicitud HTTP a otro Service general denominado HTTP-Service, el cual fue desarrollado para centralizar esta tarea y encargarse de interpretar las respuestas del Backend realizando el proceso de deserialización de los datos, manejo de errores y notificación al servicio que lo invocó del resultado de la petición.

HttpService
- baseUrl: String
- methods: List<Methods>
- defaultOptions: HttpOptions
+ get(String path, HttpOptions options)
+ post(String path, Object data, HttpOptions options)
+ put(String path, Object data, HttpOptions options)
+ delete(String path, Object data, HttpOptions options)
- doRequest(String path, Object data, HttpOptions options)
- getAuthToken(): String
- processUri(PathTemplate path): String

Figura 7.27: estructura del módulo de servicio HTTP-Service

En la figura [7.27](#) podemos ver la estructura de la clase HTTP-Service, el cual modela los métodos HTTP como métodos de instancia (funciones get, post, put y delete) los cuales reciben por parámetro la URI del endpoint para realizar la solicitud, así como los datos a incluir y opcionalmente un objeto con opciones, entre ellas encabezados HTTP, campos a incluir en la respuesta, tipo de feedback mostrado al usuario, etc. Este cliente HTTP se inicializa en el momento de carga con un baseUrl el cual representa la URL base del servidor donde se encuentra desplegado Transcriptor-Backend.

La autenticación de las solicitudes también es contemplada por este módulo, enviando el token de autenticación solicitado por la API a través del encabezado HTTP Authorization, el cual se configura durante el inicio de sesión del usuario.

Luego, como se mencionó. cada servicio delega la lógica de solicitud a la API utilizando este cliente según el siguiente ejemplo:

```

1 private loginPath = '/api/login';
2 login(loginCredentials: LoginCredentials) {
3     return this.httpService.post(this.loginPath, loginCredentials);
4 }

```

Figura 7.28: Método login, en el servicio LoginService, ejemplo de delegación de respuesta al HTTP-Service.

En la figura [7.27](#) se puede observar un fragmento de la clase LoginService, que es el módulo de servicio encargado de realizar la solicitud al endpoint de inicio de sesión (/login). En el cuerpo de la función login se realiza la delegación de la lógica de solicitud al HTTP-Service, enviando por parámetro la URI relativa al endpoint de la API y las credenciales de autenticación. Lo que retorna un objeto de tipo Observable, el cual permite definir una función a ejecutar cuando se complete la solicitud HTTP con su correspondiente respuesta.

7.5.4. Secciones

Transcriptor-Frontend esta conformado por varias pantallas, las cuales permiten realizar diferentes acciones, tanto de colaboración, como de navegación a través de los proyectos de transcripción sobre los que esta trabajando la comunidad. A continuación realizaremos un recorrido por las diferentes vistas de la aplicación describiendo su funcionamiento y/o la información que brindan.

Portada (Landing Page)

La portada es la pagina inicial o pagina de aterrizaje (landing page) que el usuario visualiza cuando ingresa al sitio, le brinda información las sobre principales características de la plataforma y permite acceder a iniciar sesión o registrarse. (ver figura [7.29](#))

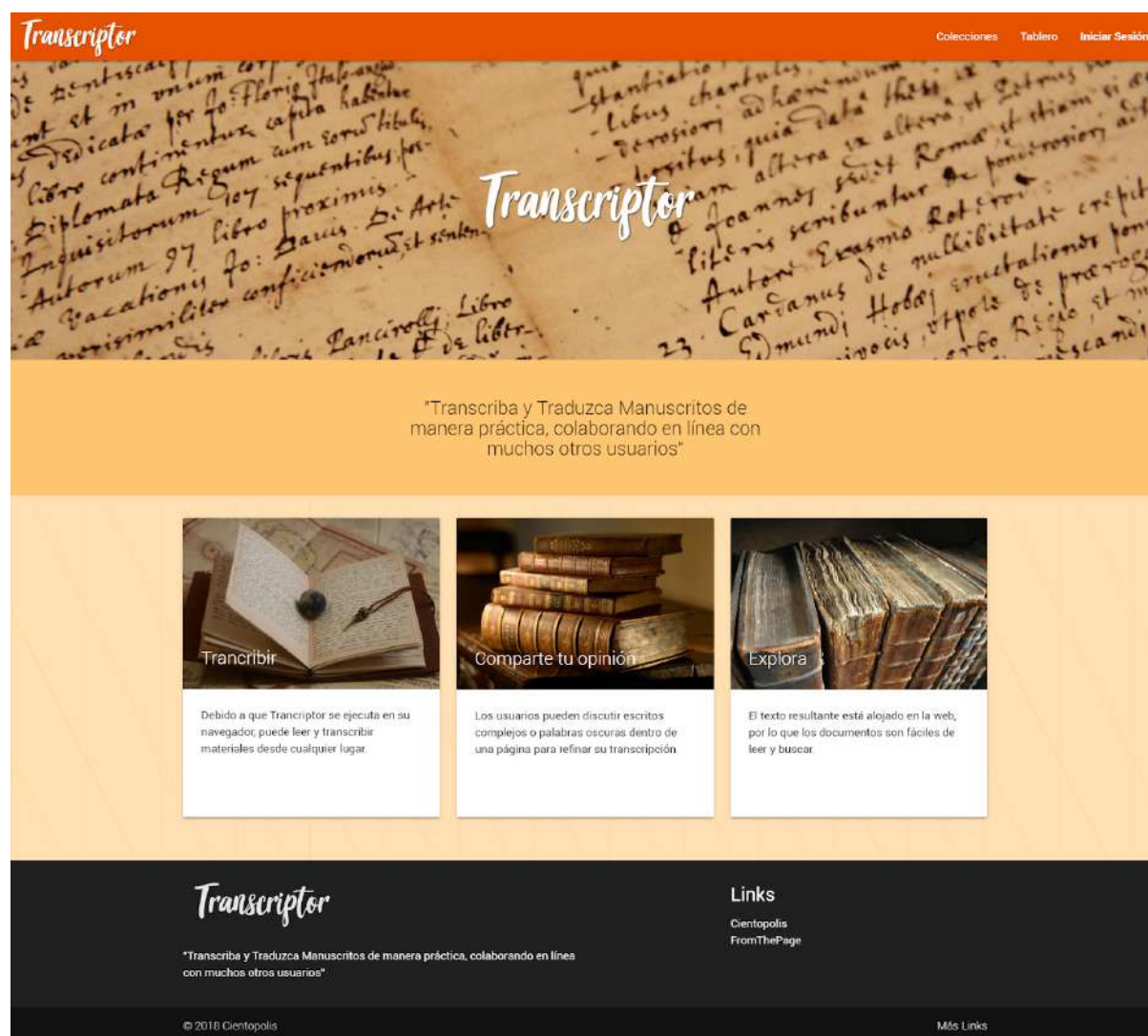


Figura 7.29: Portada de la plataforma.

Pantalla Principal (Home)

Una vez autenticado el usuario será redirigido a la pantalla principal de la plataforma, donde va a poder visualizar acontecimientos recientes dentro del sistema e información relevante dependiendo el perfil que posea, de la siguiente manera:

- **Usuario Administrador:** Puede visualizar las estadísticas de la plataforma, cantidad de colecciones, trabajos y paginas transcriptas. Además se le brindará la posibilidad de crear nuevas colecciones y proyectos de transcripción en la plataforma.
- **Usuario Propietario:** Puede visualizar las estadísticas de sus colecciones y trabajos asignados. Si el usuario es administrador de alguna colección se le permitirá cargar nuevos proyectos en la misma.
- **Usuario Colaborador:** Se le informan sus estadísticas sobre sus aportes a la comunidad, así como el rango de colaborador que posee en el juego. Puede ver además, los proyectos en los cuales se encuentra trabajando actualmente para un fácil acceso y reincorporación al trabajo que realizaba.

The screenshot shows the Transcriptor home page for an administrator. The top navigation bar includes the logo 'Transcriptor', the text 'Colecciones Tablero', and a user profile icon for 'administrador' with the role 'Visitador'. The main content area is titled 'Tablero' and features a 'TUS TRABAJOS' section with a 'INICIAR UN PROYECTO' button. Below this, there are several collection cards: 'collection title', 'collection2', 'collection3', 'collection4', 'Cuentos Argentinos', and 'Textos'. Each card displays the collection name and a message: '¡Aún no existen Trabajos en esta Colección!'. On the right side, there are two summary widgets: 'Estado del Sitio' showing 6 Collections, 1 Trabajo, and 0 Notas; and 'Tu Actividad' showing a list of recent actions performed by the administrator, such as editing pages and transcribing text. The footer contains the Transcriptor logo, a tagline: '*Transcribe y Traduzca Manuscritos de manera práctica, colaborando en línea con muchos otros usuarios*', copyright information '© 2018 Cientopolis', and a 'Links' section with 'Cientopolis' and 'FromThePage'.

Figura 7.30: Pantalla principal de transcriptor con rol Administrador.

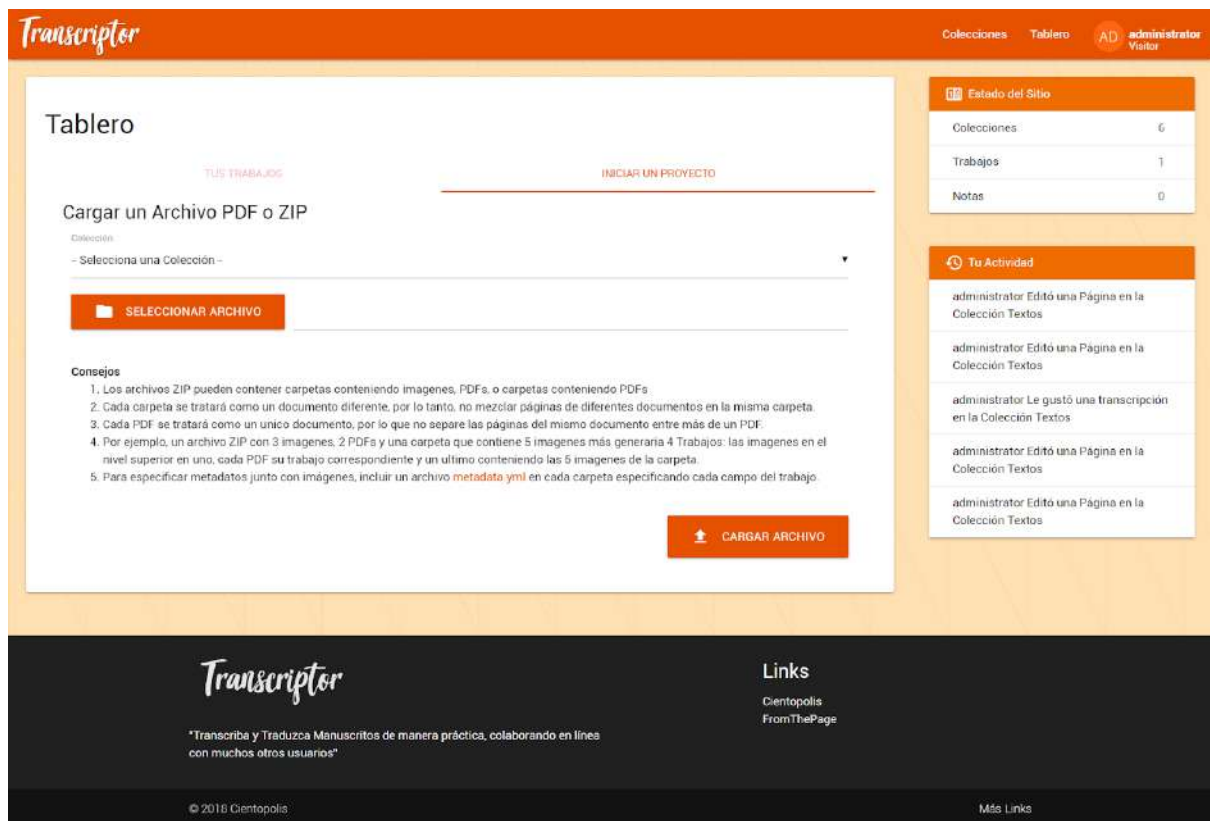


Figura 7.31: Pantalla principal de transcriptor con rol Administrador, con opciones de carga de nuevas colecciones.

Colecciones

Esta sección permite explorar las colecciones cargadas con proyectos disponibles para transcribir. En el caso de poseer el rol de administrador muestra la opción para acceder al formulario de creación de una nueva colección (ver figura [7.32](#)).

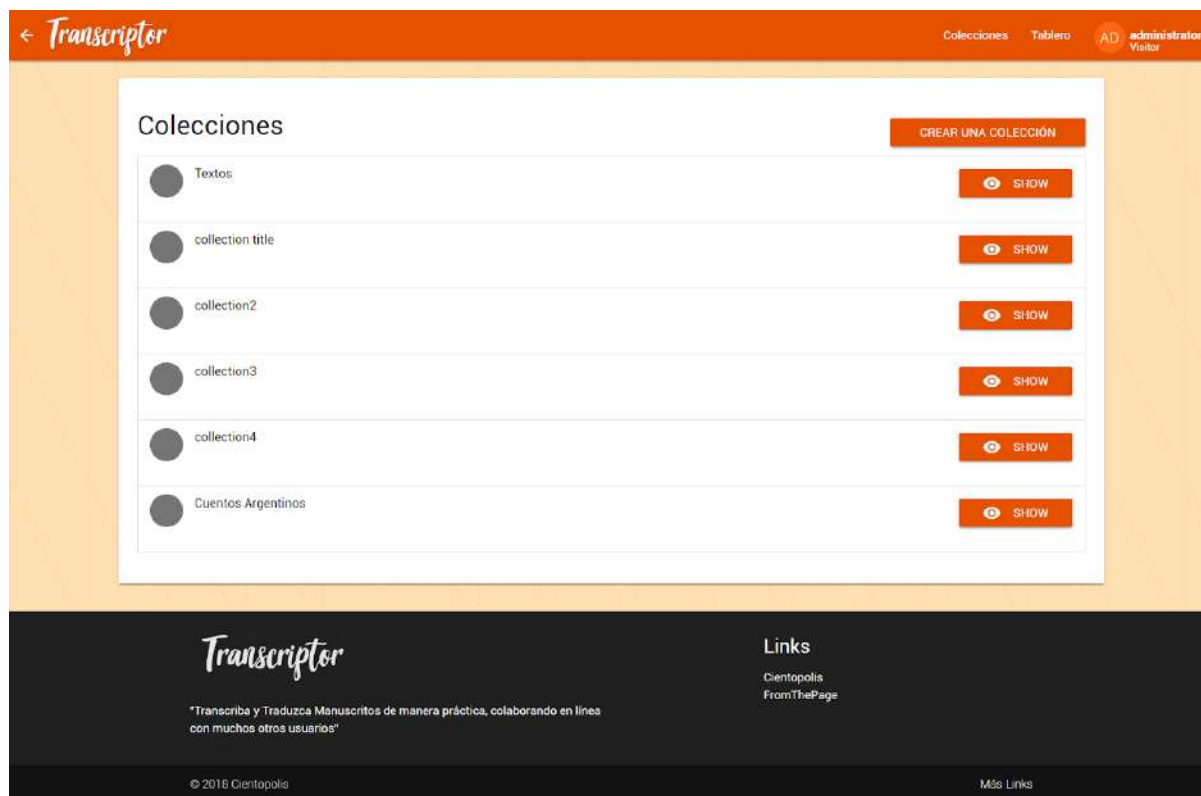


Figura 7.32: Listado de colecciones disponibles.

Al acceder al detalle de una colección se muestra una pantalla (ver figura [7.33](#)) donde se examina el contenido de la misma, es decir, los proyectos que agrupa. Desde esta sección, en el caso de tener permisos para administrarla, también se puede cambiar la configuración de la colección seleccionada, dando la posibilidad de agregar detalles históricos, indicaciones para el proceso de transcripción y eliminar la colección (ver figura [7.34](#)).

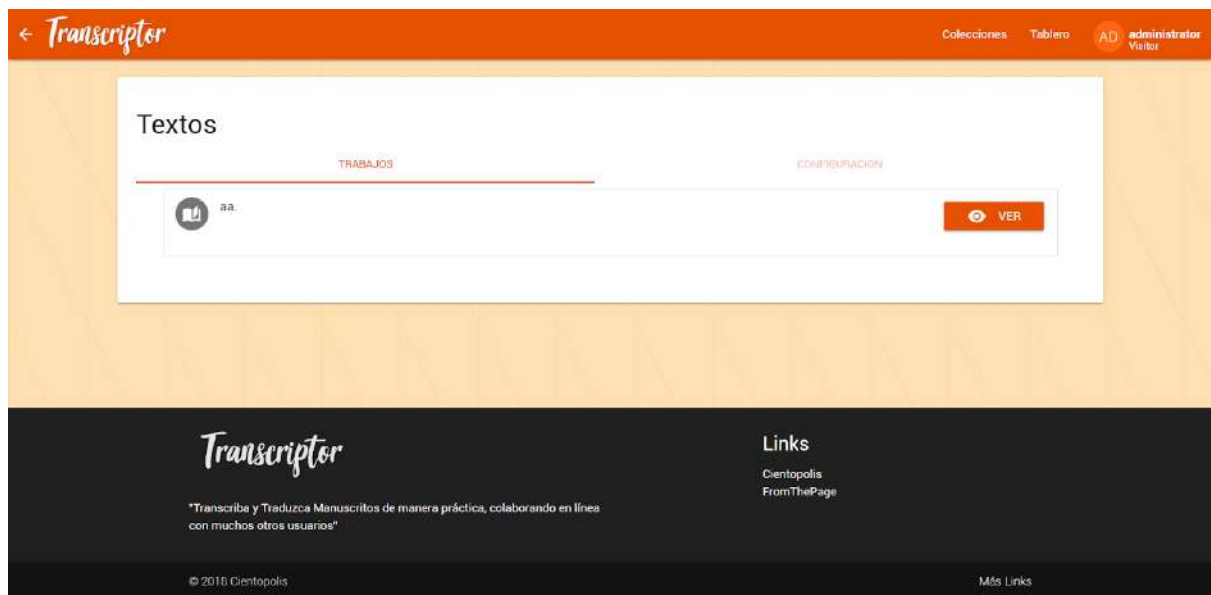


Figura 7.33: Listado de proyectos asociados a la colección.

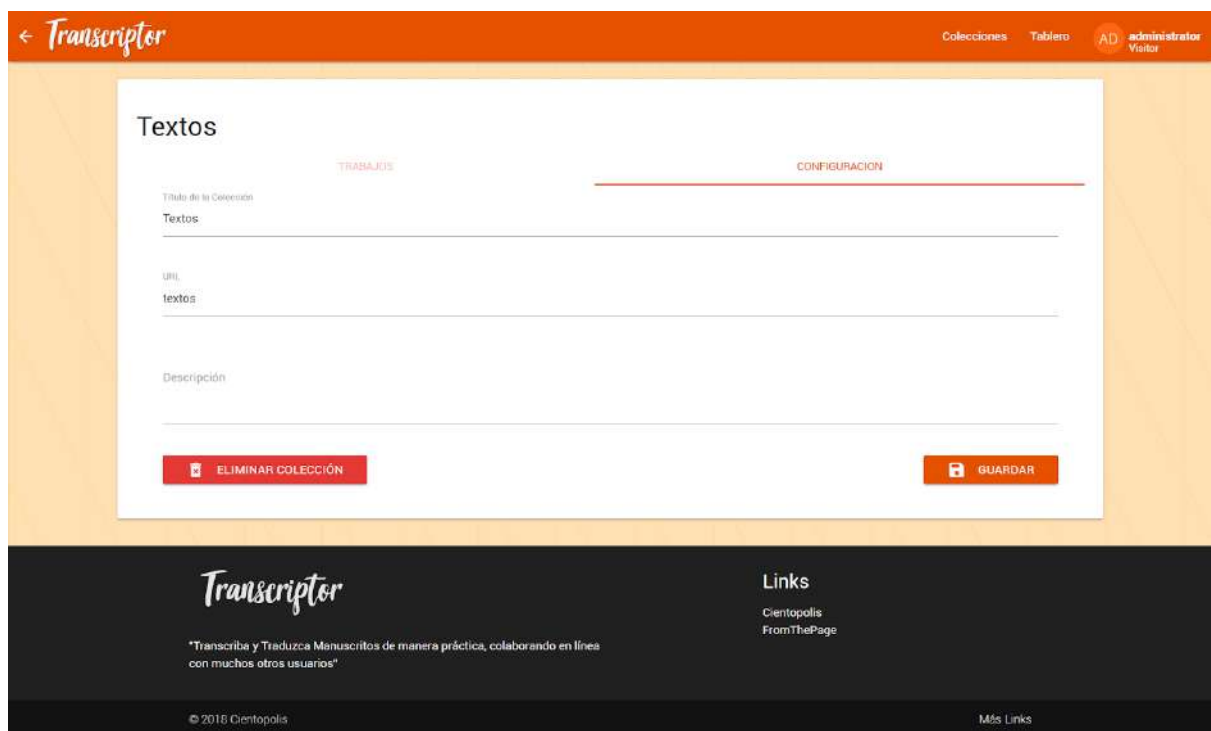


Figura 7.34: Pantalla de configuración de la colección.

Proyecto

De la misma manera que las colecciones, el detalle de un proyecto permite realizar cambios en su configuración donde es posible agregar datos de autor, año de publicación del manuscrito y su historia (figura 7.35). Además, la pantalla administra las componentes asociadas al proyecto, es decir, las paginas a transcribir permitiendo listarlas, borrarlas o cargar nuevas para que estén disponibles para su transcripción.

The screenshot displays the 'Transcriptor' web application interface. At the top, there is a navigation bar with the 'Transcriptor' logo on the left and user information 'AD administrator Visitor' on the right. The main content area is divided into two sections: 'PAGINAS' and 'CONFIGURACION'. Under 'PAGINAS', there is a list of three items, each with a document icon, a number (1, 2, 3), and a 'TRANSCRIPCIÓN' button. The first item is labeled 'transcribed'. To the right, a sidebar titled 'Actividades' shows a list of recent actions performed by the administrator, such as 'Editó una Página en el Trabajo aa' and 'Le gustó una transcripción en el Trabajo aa'. The footer contains the 'Transcriptor' logo, a tagline '*Transcriba y Traduzca Manuscritos de manera práctica, colaborando en línea con muchos otros usuarios*', copyright information '© 2018 Cientopolis', and a 'Links' section with 'Cientopolis' and 'FromThePage'.

Figura 7.35: Detalle de proyecto con listado de páginas.

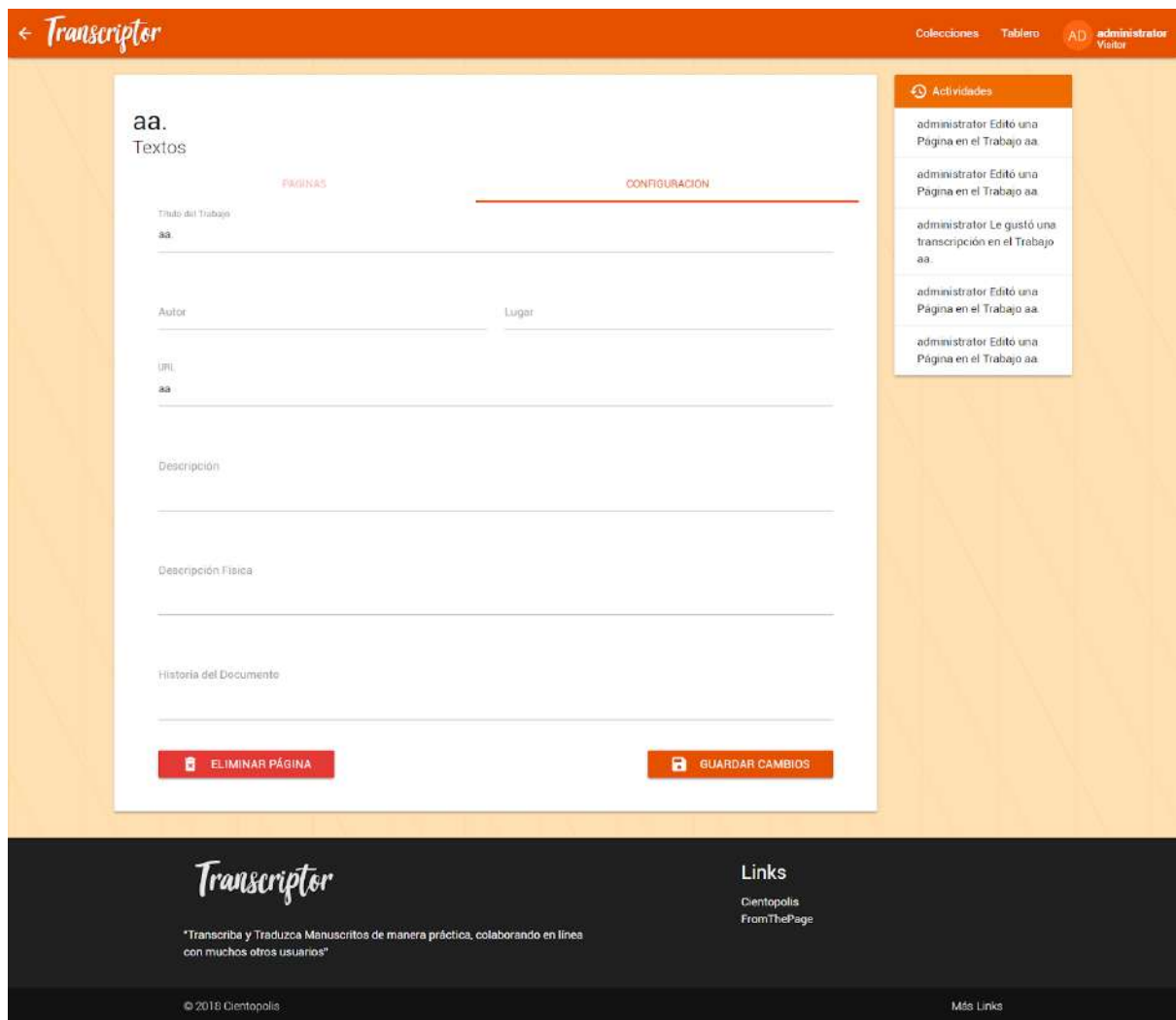


Figura 7.36: Configuración del proyecto.

Perfil de usuario

Esta pantalla permite visualizar al usuario sus datos de perfil, permitiendo editarlos en caso que sea necesario. Además, en esta vista podrá consultar el listado de insignias obtenidas y su rango en el juego de transcripción (figura 7.37).

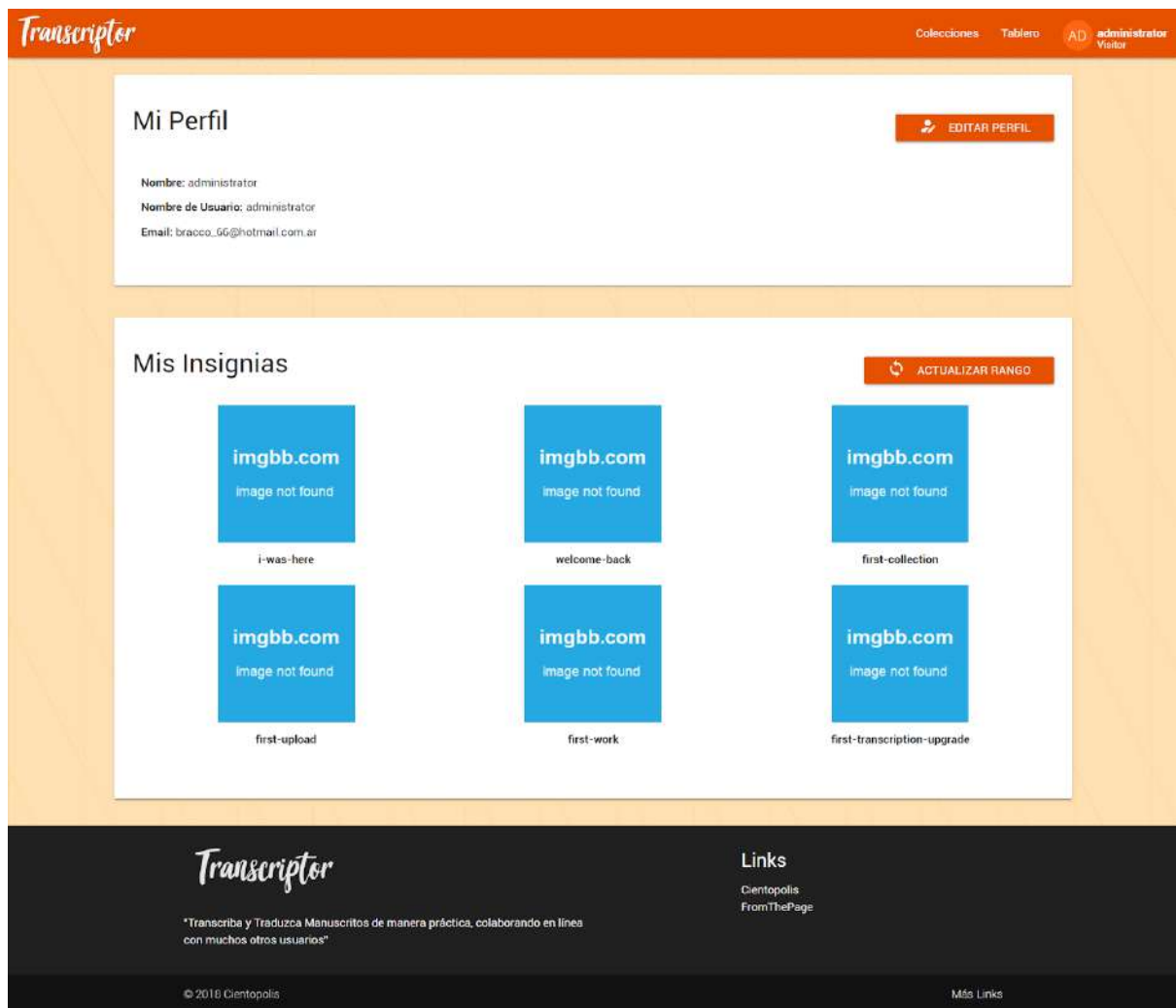


Figura 7.37: Perfil e insignias del usuario.

Finalmente solo queda una última vista a presentar, la cual es la mas importante por su lógica, cantidad de flujos y funcionalidad que brinda a la plataforma y a la se dedicará el siguiente inciso de este capítulo.

Entorno de Transcripción

El entorno de Transcripción, o pantalla de transcripción de un manuscrito, es la sección principal dentro del Transcriptor-Frontend, esto se debe a que es aquí donde se materializa y se genera la información de las transcripciones de manera colaborativa y donde se cumple el flujo de transcripción planteado por la plataforma.

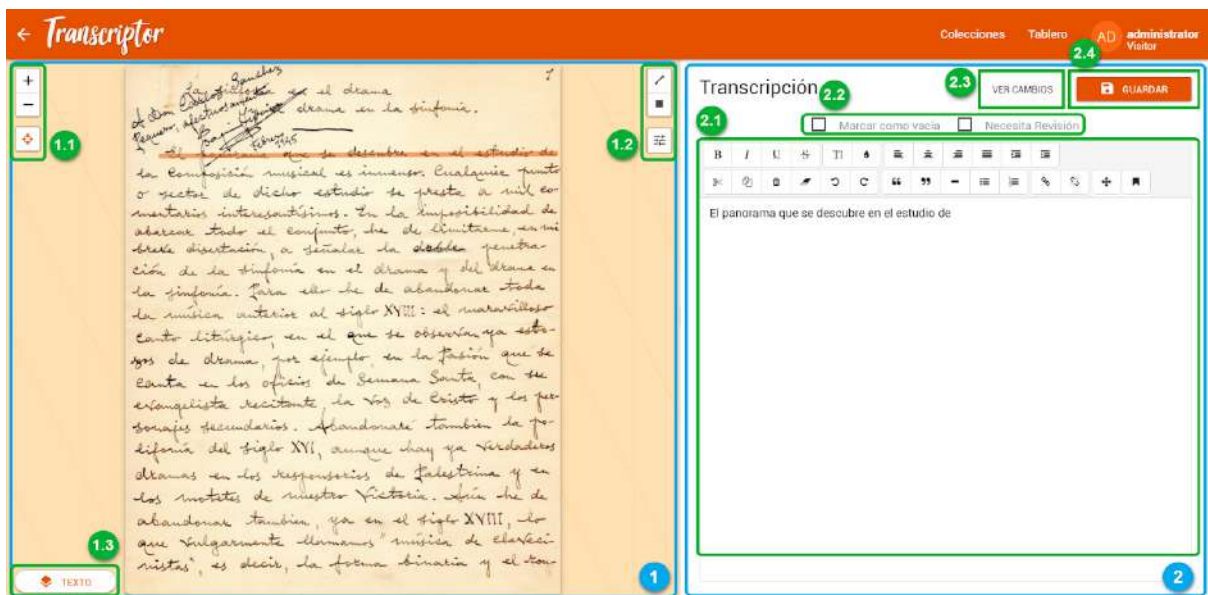


Figura 7.38: Entorno de transcripción.

Al ingresar a esta pantalla se distinguen varios componentes (ver figura 7.38):

- **Visor de documento:** permite visualizar el documento digitalizado, al estar planteado utilizando la librería de mapas Leaflet y utilizando como capa principal la imagen del manuscrito, es posible realizar acercamientos a los distintos fragmentos del mismo, navegarlo y hasta realizar marcas de manera simple delegando la lógica a la librería (ver en figura 7.38, anotación 1).

 - **Controles de Zoom:** permiten controlar el nivel de acercamiento sobre el documento (ver en figura 7.38, anotación 1.1).
 - **Controles de Dibujo:** permiten entrar en modo de marcado, utilizando Leaflet Draw logra dibujar polígonos sobre las áreas a transcribir (ver en figura 7.38, anotación 1.2).
 - **Ajustes de imagen:** habilita controles para cambiar el brillo, contraste y saturación de la imagen digitalizada para facilitar la visualización del texto escrito (ver en figura 7.38, anotación 1.3).
 - **Selector de capa:** abre el menú de selección de capas de transcripción, desde este menú también es posible acceder a la pantalla de administración de las mismas, que permite modificarlas o eliminarlas (ver en figura 7.38, anotación 1.4).
- **Panel de Transcripción:** Contiene la lógica de interacción del usuario con la el estado de la transcripción (ver en figura 7.38, anotación 2).

 - **Editor de texto:** componente destinada al ingreso y previsualización del texto final, permite seleccionar regiones del mismo para referenciarlas a secciones del documento. Al tratarse de un editor de texto WYSIWYG posee funciones para aplicar estilos al texto, negrita, subrayado, entre otros (ver en figura 7.38, anotación 2.1).

- **Botón de guardado:** ejecuta el guardado del estado actual del texto transcrito (ver en figura 7.38, anotación 2.2). Botón “Ver Cambios”: acceso al menú de cambios de la transcripción del manuscrito (ver en figura 7.38, anotación 2.3).
- **Selectores de estado:** permite marcar la pagina en estado “vacía” o “pendiente de revisión” (ver en figura 7.38, anotación 2.4).

De esta manera, una vez presentados los componentes que integran la sección es posible ahondar en los diferentes elementos y flujos que existen en esta sección, los cuales están planteados para facilitar al usuario la tarea de transcripción y a su vez lograr representar de una forma computable los datos contenidos en los manuscritos, manteniendo la noción de colaboración.

7.5.5. Elementos de transcripción y generación de texto

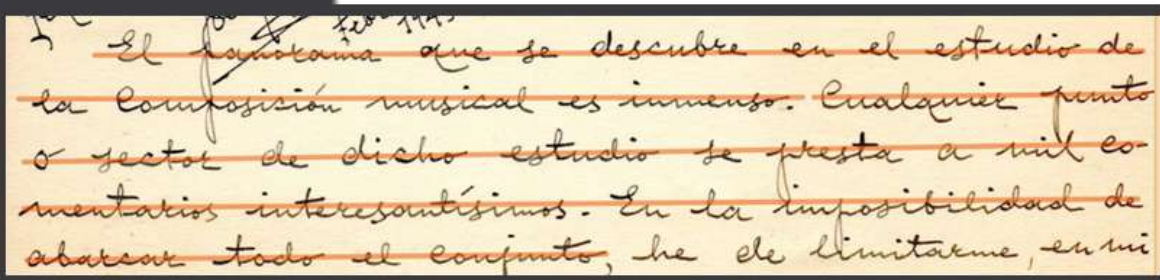
Transcriptor modela las transcripciones utilizando relaciones entre sus entidades, buscando no sólo representar el contenido textual de un manuscrito, sino también enriquecerlo asociándolo directamente al contexto en el que se encuentra plasmado en el papel, de esta manera se habla de mantener una referencia entre fragmentos del documento y texto transcrito.

Las Marcas (Mark) son entidades que modelan una línea o polígono la cual selecciona una porción del documento digitalizado, poseen todos los elementos necesarios para que el cliente pueda renderizarla sobre el manuscrito, entre ellos, un conjunto de coordenadas relativas a su imagen digital, las cuales son utilizadas para posicionarse y dibujarse en el visor.

En cuanto al texto se manejan dos representaciones, una directamente ligada a la marca, a la cual denominamos Transcripción (Transcription), que describe el contenido textual transcrito por el usuario para una marca sobre el documento, y otra representación que se encuentra orientada a mantener la estructura y progreso total del proceso de transcripción llamada Plantilla de Transcripción (Transcription Template).

Las Plantillas de transcripción son “Mapas” de relaciones los cuales se utilizan para referenciar el orden en el cual el texto de las marcas se combina, permitiendo generar dinámicamente el texto completo del documento a transcribir.

Marcas sobre el documento



Plantilla de Transcripción

```
1 <span class="contribution-mark-13">{{ contributionMark13 }}</span>
2 <span class="contribution-mark-14">{{ contributionMark14 }}</span>
3 <span class="contribution-mark-15">{{ contributionMark15 }}</span>
4 <span class="contribution-mark-16">{{ contributionMark16 }}</span>
5 <span class="contribution-mark-17">{{ contributionMark17 }}</span>
6 <span class="contribution-mark-18">{{ contributionMark18 }}</span>
```

Plantilla con reemplazos dinámicos

```
1 <span class="contribution-mark-13">El panorama que se descubre en el estudio de</span>
2 <span class="contribution-mark-14">la composición musical es inmenso.</span>
3 <span class="contribution-mark-15">Cualquier punto</span>
4 <span class="contribution-mark-16">o sector de dicho estudio se presta a mil co</span>
5 <span class="contribution-mark-17">mentarios interesantísimos. En la imposibilidad de</span>
6 <span class="contribution-mark-18">abarcarse todo el conjunto,</span>
```

Texto Generado

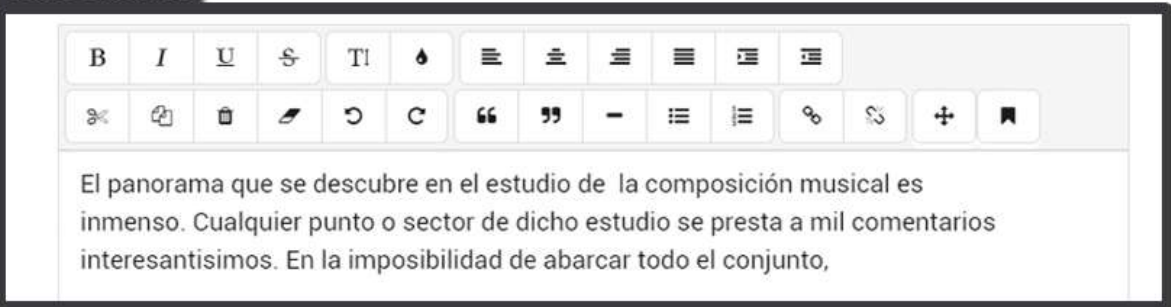


Figura 7.39: Generación de transcripción en base a las marcas del documento.

Como se observa en la figura [7.39](#), para generar el texto, Transcriptor utiliza las marcas que el usuario realiza sobre el documento y, a medida que estas se van agregando, Transcriptor-Backend se encarga de concatenarlas en el texto completo utilizando una referencia a la marca que ocupa ese espacio, formando la plantilla de transcripción (ver figura [7.39](#) recuadro “Plantilla de Transcripción”). Luego, al iniciar el entorno de transcripción Transcriptor-Frontend utiliza la plantilla como guía para reemplazar las referencias por el texto de cada marca indicada (figura [7.39](#) recuadro “Plantilla con reemplazos dinámicos”) y finalmente renderiza dicho contenido en el editor de texto (figura [7.39](#) recuadro “Texto generado”), donde podrán visualizarse las asociaciones creadas.

7.5.6. Estrategias de transcripción

Para generar la transcripción completa del manuscrito, Transcriptor implementa múltiples estrategias para lograrlo, brindando al usuario varios caminos para organizar su propia lógica de trabajo manteniendo el mismo objetivo.

En general las estrategias de transcripción están compuestas por tres procesos:

- **Proceso de marcado:** donde el transcriptor selecciona un fragmento del documento digitalizado utilizando una línea o un polígono para “marcarlo”, de esta manera es posible identificarla y relacionarla posicionalmente con una transcripción.
- **Proceso de carga:** proceso por el cual el usuario transcribe el texto contenido en el documento digitalizado, ya sea de manera total o parcial.
- **Proceso de guardado:** implementa la lógica para persistir la transcripción creada, al prescindir de la intervención del usuario este proceso es transparente desde su punto de vista.

Si bien estos procesos son independientes uno del otro, el orden en el que se ejecutan cambia la manera en la que el usuario los realiza. Las estrategias de transcripción soportadas por Transcriptor son:

- **Marcar para Cargar:** el usuario selecciona una fracción del documento digitalizado generando una marca para luego relacionarla con una nueva transcripción.
- **Seleccionar para Marcar:** desde el editor de texto, seleccionar parte del texto y relacionarlo con una marca generada en el momento.
- **Transcripción Alternativa:** en esta estrategia, el usuario propone una nueva transcripción para una relación marca-transcripción ya existente, a manera de corrección sobre su contenido.
- **Transcripción Semántica:** mantiene el mismo principio que el método “marcar para cargar”, solo que la marca se relaciona a una Transcripción Semántica, descrita en próximos incisos.

A continuación se describe de manera detallada cada uno de estas estrategias de transcripción, haciendo énfasis en los pasos a seguir para realizar cada una de estas en Transcriptor. La estrategia “Transcripción semántica” será abordada más adelante una vez introducidos los conceptos de semántica en transcripciones.

Estrategia de transcripción: Marcar para Cargar

La estrategia de transcripción “Marcar para Cargar” describe el flujo que, como indica su nombre, esta dado por la ejecución del proceso de marcado en una primera instancia, para luego cargar la información textual sobre la marca que se acaba de realizar.



Figura 7.40: Selección de la herramienta de marcado.

El proceso de marcado inicia al momento de seleccionar una herramienta de selección, dando la posibilidad de optar entre una marca lineal o cuadrada (ver figura 7.40). A continuación el visor ingresa al modo de marcado.

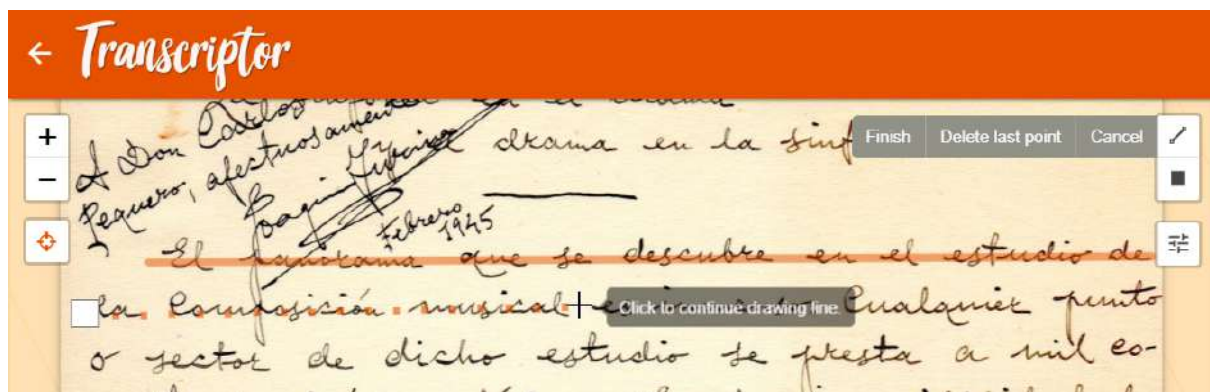


Figura 7.41: Creación de marca sobre línea del manuscrito.

Durante el modo de marcado, se habilita al usuario la posibilidad de interactuar con el visor del documento para dibujar la marca sobre el fragmento a transcribir. Como se puede observar en la figura 7.41, el colaborador realiza la selección sobre la línea de texto del documento que se dispone a transcribir.

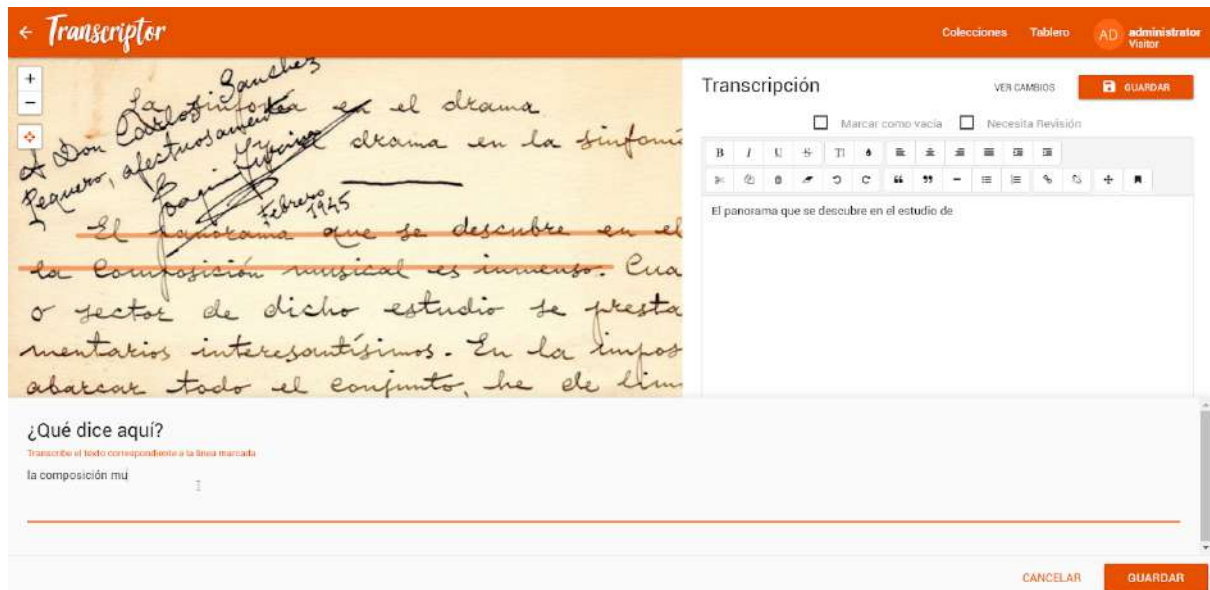


Figura 7.42: Ventana emergente durante el proceso de carga.

Una vez creada la marca, se inicia el proceso de carga. Para esta estrategia el proceso de carga esta implementado utilizando una ventana emergente (modal), la cual posee un campo de texto donde el usuario fácilmente puede ingresar el texto de la transcripción. Finalmente, al seleccionar el botón “Guardar” se desencadena el proceso de guardado de la marca.

```

1 private createPath = '/api/mark';
2 create(mark: Mark): Observable<Mark> {
3     return this.httpService.post(this.createPath, mark) as Observable<Mark>;
4 }

```

Figura 7.43: Método de creación de marcas en el mark-service.

```

1  export class Mark {
2      id:number;
3      transcription_text:string;
4      shape_type:string;
5      coordinates:LatLng[];
6      page_id:number;
7      layer_id: number;
8
9      constructor(page,layer,layerType,transcriptorLayer){
10         this.page_id = page.id;
11         this.layer_id = transcriptorLayer.id;
12         this.coordinates = layer.getLatLngs();
13         this.shape_type = layerType;
14     }
15 }

```

Figura 7.44: Definición de la clase Marca (Mark).

Durante el proceso de guardado, entre los distintos componentes, interviene el módulo servicio de marcas (mark-service) que se encarga de realizar la solicitud y procesar la respuesta de los endpoints concernientes a las marcas en Transcriptor-API, utilizando el cliente HTTP desarrollado para realizar los llamados a la API. Para este flujo se hace uso del método de creación de marcas (ver fig. 7.43) que recibe por parámetro un objeto de tipo Marca (Mark) para requerir la persistencia de la marca en el backend.

La clase Marca (ver fig. 7.44) posee los atributos básicos para la creación del conjunto marca-transcripción. Entre los principales atributos para el proceso de carga se encuentran:

- **transcription_text:** Representa el texto a guardar en una transcripción, la cual será asociada a la marca a crear.
- **shape_type:** Indica el tipo de polígono dibujado sobre el documento, sus valores pueden ser: Polyline o Rectangle.
- **coordinates:** Colección de coordenadas utilizadas para realizar el dibujo de la marca sobre el documento.
- **page_id:** El identificador de la pagina a la cual la marca esta asociada.
- **layer_id:** El identificador de capa a la cual la marca esta asociada.

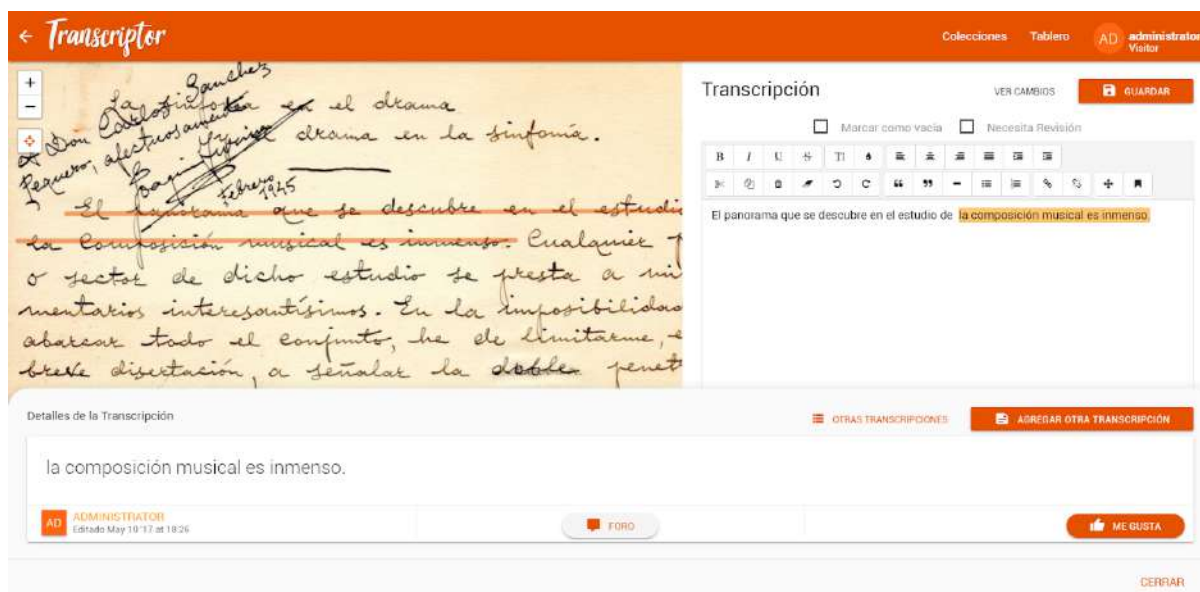


Figura 7.45: Visualización de relación entre marca y transcripción.

Como resultado final, podemos observar en la figura 7.45 como el fragmento del texto transcrito queda relacionado con la marca sobre el documento digitalizado, de manera que es posible seleccionar cualquiera de estos dos para poder obtener una referencia posicional dentro del contexto del documento.

Estrategia de transcripción: Seleccionar para Cargar

De manera inversa a la estrategia Marcar para cargar, la estrategia “Seleccionar para Marcar” permite generar marcas a partir de texto cargado en el editor de texto del panel de transcripción. Esto permite mayor fluidez y control sobre el texto y el proceso, ya que el usuario puede optar en qué momento generar la marca del texto que acaba de escribir, sin necesidad de realizar el proceso completo cada vez que transcribe un fragmento del documento.

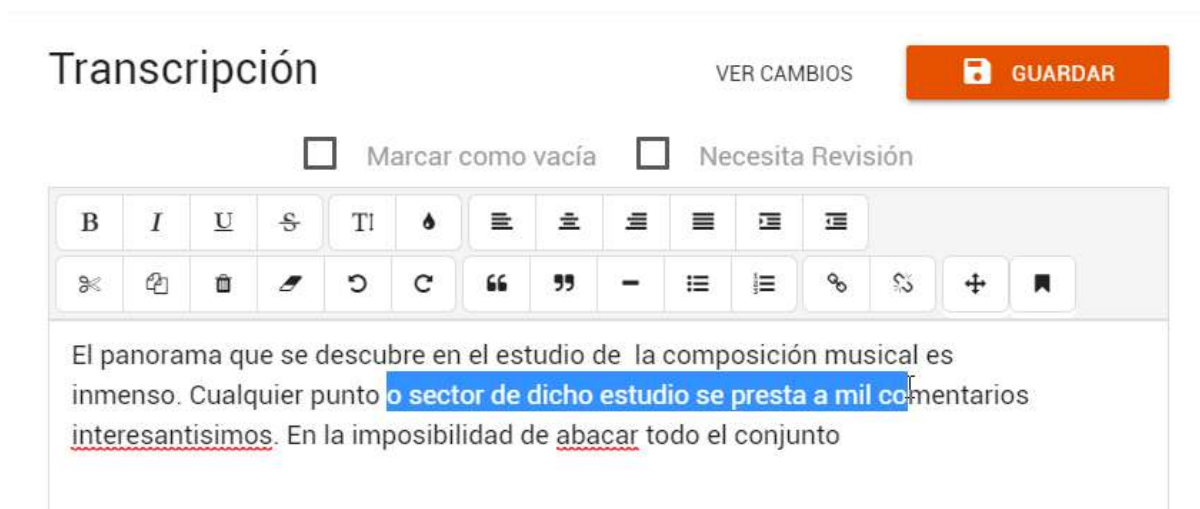


Figura 7.46: Selección del texto a marcar en el documento.

La estrategia inicia con el proceso de carga sobre el editor de texto, de esta manera, el usuario transcribirá un fragmento elegido volcando todo el contenido en el editor. Luego, para poder generar las marcas, el usuario selecciona el texto con su cursor (ver fig. 7.46) y activa la opción “Agregar marca desde texto” (ver fig. 7.47).

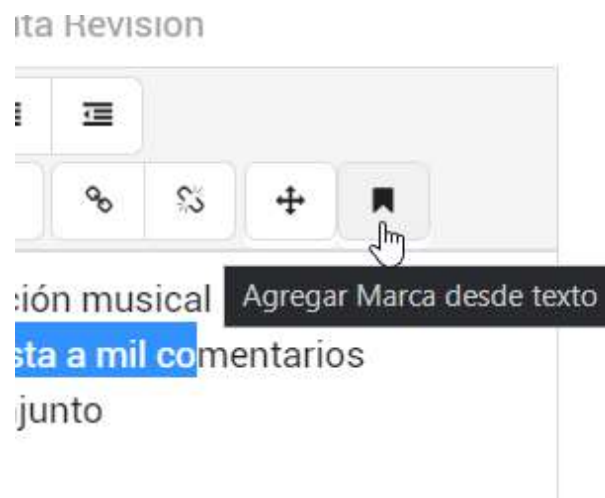


Figura 7.47: Opción “Agregar marca desde texto”.

A partir de este momento, se le solicita al colaborador que seleccione una herramienta de marcado para iniciar este proceso (ver fig. 7.48).

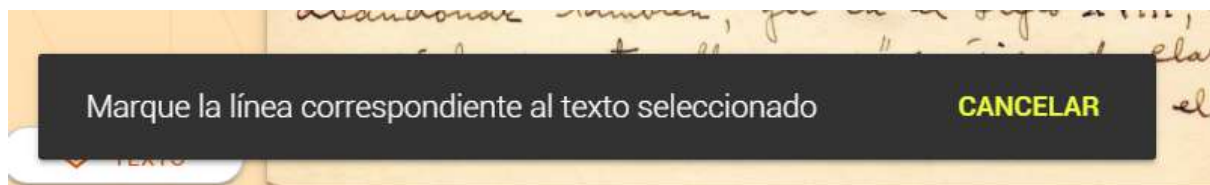


Figura 7.48: Notificación informativa para iniciar el proceso de selección.

Como ultimo paso del flujo, el usuario dibuja la marca utilizando la herramienta seleccionada (ver fig. 7.49), ejecutándose automáticamente el proceso de guardado para persistir la transcripción, se realiza la solicitud al método de creación en el módulo servicio de marcas (mark-service) y, de la misma manera que en la estrategia “Marcar para Cargar”, se registra la asociación texto-marca (ver fig. 7.45).



Figura 7.49: Creación de marca a partir del texto seleccionado.

Estrategia de transcripción: Transcripción Alternativa

Es posible que durante el proceso de transcripción surjan múltiples interpretaciones sobre un fragmento del documento, dando lugar a debate. Manejar estas diferentes alternativas brinda un nivel de inclusión y fomenta la colaboración en la instancia de mejora de una transcripción.

De esta manera, la estrategia “Transcripción Alternativa” da la posibilidad a los colaboradores de proponer otras interpretaciones a la que otro usuario generó inicialmente.

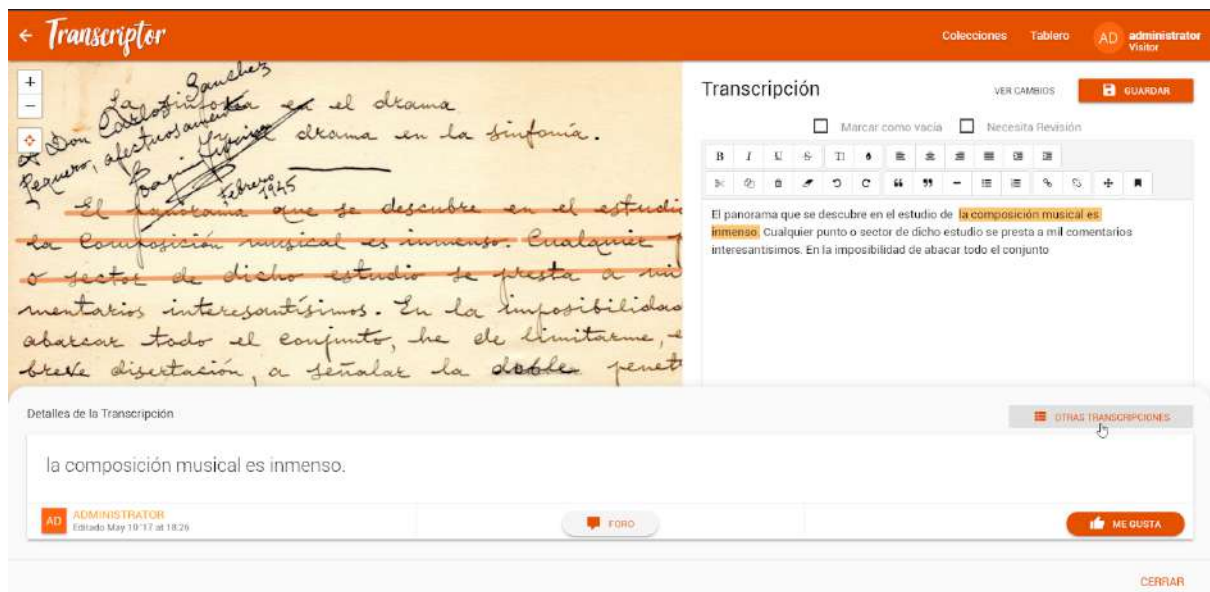


Figura 7.50: Opción de listado de transcripciones alternativas en el detalle de la marca.

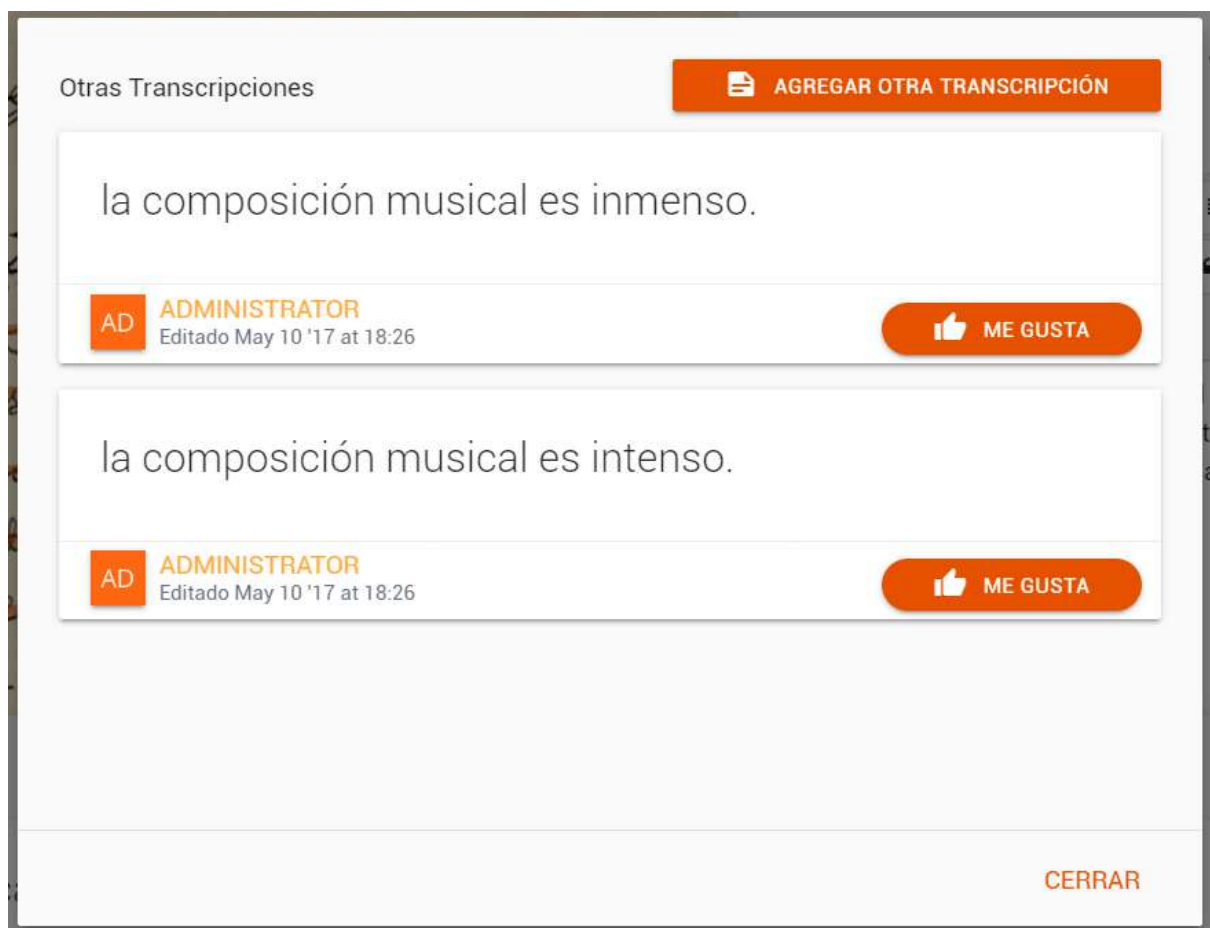


Figura 7.51: Listado de transcripciones propuestas.

Partiendo del detalle de una marca, la plataforma permite listar todas las transcripcio-

nes sugeridas por los usuarios (ver fig. 7.50 y 7.51) y permite crear una nueva transcripción alternativa a la actual.

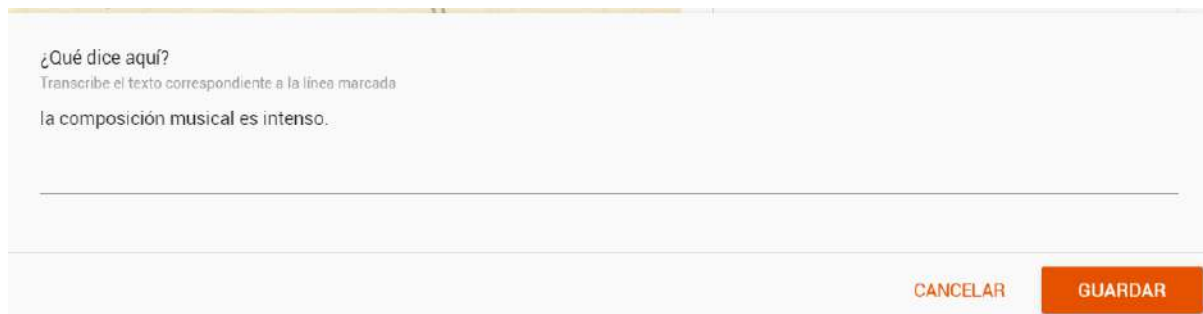


Figura 7.52: Formulario de creación para la nueva transcripción.

Al ingresar a la opción “Agregar otra transcripción” se inicia el proceso de carga a través de una ventana emergente, donde se le da la posibilidad al usuario para cargar el contenido de la nueva transcripción (ver fig. 7.52). Finalmente cuando el usuario confirma la carga se inicia el proceso de guardado, esta vez haciendo referencia al método de creación en el Módulo servicio de transcripciones (transcription-service), el cual realiza la solicitud enviando los datos de la nueva transcripción al endpoint de creación de transcripciones, donde se persiste y se agrega a la colección de contribuciones alternativas de la marca.

```
1 private createPath = '/api/transcription';
2 create(transcription, options = {}) {
3     return this.httpService.post(this.createPath, transcription, options);
4 }
```

Figura 7.53: Método de creación de transcripciones en el transcription-service.

Para tener en cuenta:

- Utilizando esta estrategia no es necesario realizar el proceso de marcado, esto se debe a que se está proponiendo contenido nuevo a una marca ya existente.
- Estas transcripciones alternativas tienen la posibilidad de convertirse en la transcripción principal de la marca, siguiendo un proceso de selección presentado en el próximo inciso.

7.5.7. Competencia de transcripciones

La estrategia de transcripción “Transcripción alternativa” (presentada en la sección anterior), generar un marco de propuesta de transcripciones alternas, que pueden lograr representar de una manera más acertada el contenido de un fragmento marcado en un documento.

Como ya mencionamos, cada marca posee un listado de transcripciones propuestas por los usuarios (fig. 7.51) entre las cuales se encuentra la transcripción principal, la cual es la mostrada en el detalle de la marca y la que forma parte del texto final resultante del proceso.

Para realizar el proceso de selección de la transcripción principal, la plataforma utiliza el método de votación basado en likes (Me Gusta) y, en función de este valor, realiza la comparación y calculo de la transcripción principal del marca. Este proceso se describe haciendo hincapié en la implementación del servicio backend, abordado en el apartado “Mejora de transcripciones” de la sección [7.4.3](#).

De esta manera, el frontend se encarga de realizar la solicitud a lo servicios de Transcriptor-Backend utilizando la API mientras que el backend se ocupa de realizar la lógica de comparación de colaboraciones.

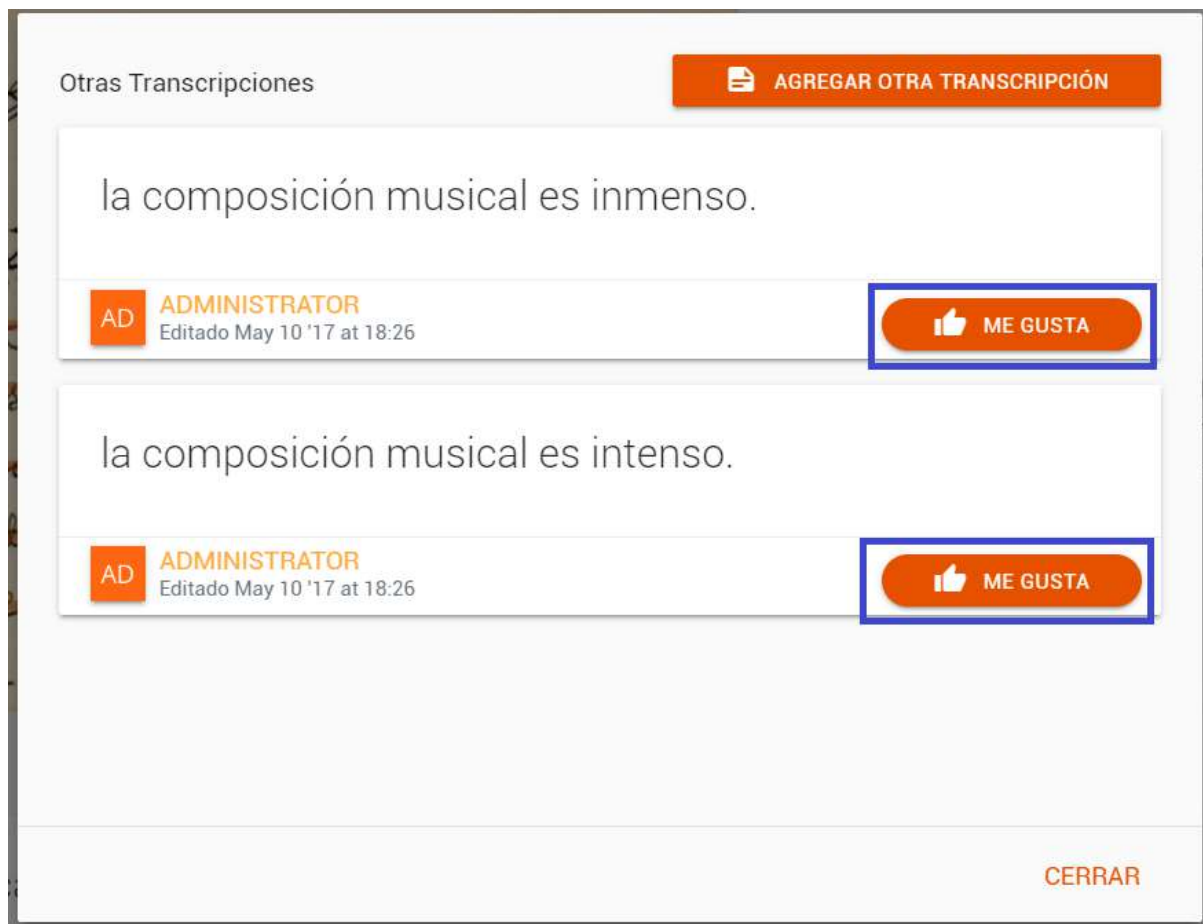


Figura 7.54: Botones de “Me Gusta”, utilizados para realizar la votación de las transcripciones.

En el momento en que el usuario vota por una transcripción, se invoca al servicio Transcription-Service quien se encarga de realizar la solicitud al servicio /transcription/like registrando el nuevo voto e inicia la competencia entre contribuciones.

La competencia entre contribuciones no es mas que un proceso comparación de transcripciones realizada en el backend utilizando la cantidad de likes recibidos por cada contribución (apartado “Mejora de transcripciones” de la sección [7.4.3](#)), en el caso superar la cantidad que posee la transcripción principal se actualiza la relación de la marca para referenciar a la “ganadora”.

Finalmente, en el frontend se consulta nuevamente la información de la marca en cuestión y, en el caso que la relación entre marca y transcripción haya cambiado, se actualizan los datos mostrados tanto en el detalle de la marca, como en el texto completo de la trans-

cripción.

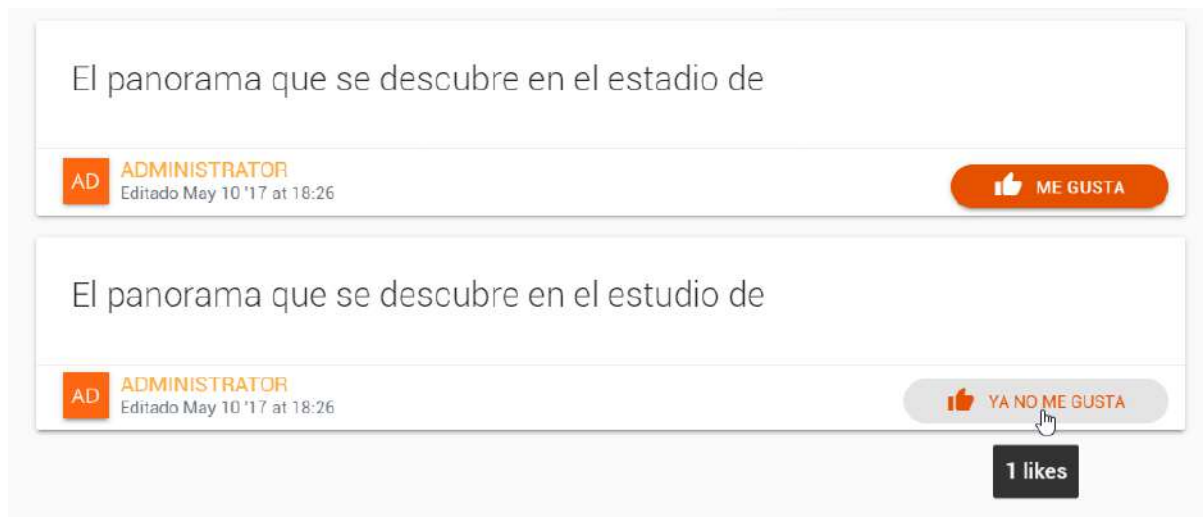


Figura 7.55: Ejemplo de competencia, se selecciona “Me Gusta” a una Transcripción.

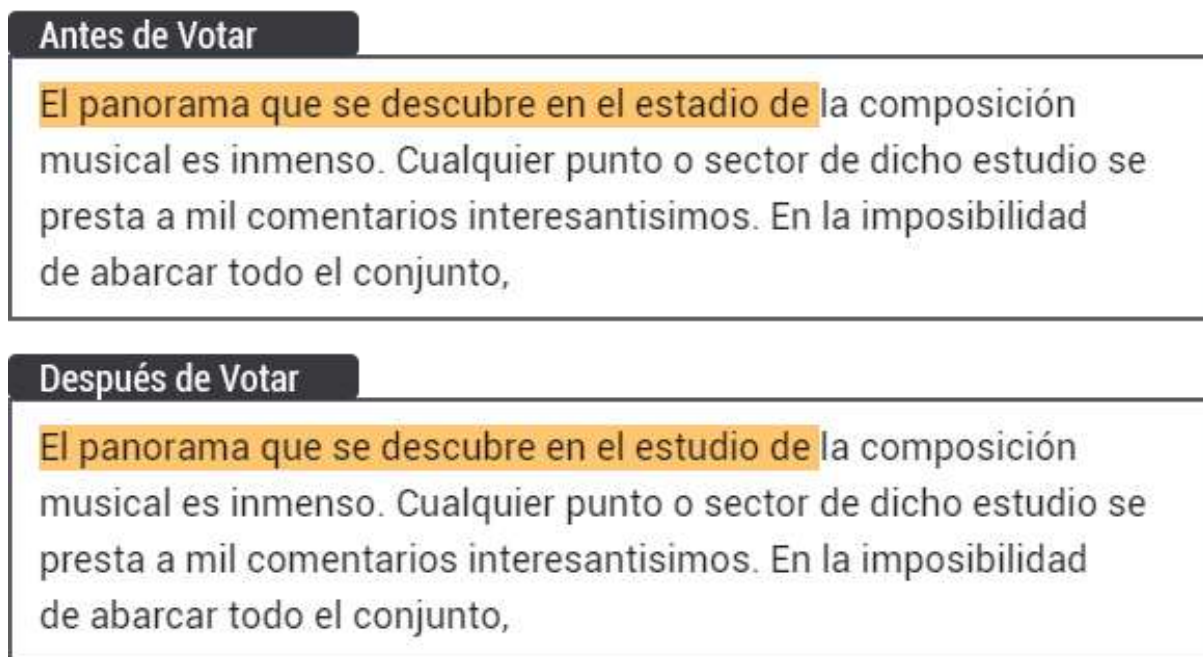


Figura 7.56: Ejemplo de competencia, contenido actualizado luego de la votación.

En la figura [7.55](#) y [7.56](#) se puede observar un ejemplo de como se realiza la actualización del texto en base a la votación de las transcripciones, la transcripción votada pasó a ser la de mayor cantidad de votos (Me gusta o likes) y empieza a formar parte del texto final de la transcripción. Luego, en el caso de que otro usuario proponga una transcripción para ésta marca y que la misma supere la cantidad de votos de la actual, pasaría a ser la nueva transcripción principal y ocupar su lugar en el texto.

Cabe mencionar que el método de actualización dinámica que posee el texto dentro del

editor esta implementada utilizando la plantilla de transcripción (template) presentada en la sección 7.5.5 de esta tesis. La cual representa la estructura del texto utilizando referencias a las marcas sobre el documento, permitiendo a Transcriptor-Frontend realizar los correspondientes reemplazos en el cuerpo del texto con el valor de las transcripciones mas votadas, de manera dinámica.

7.5.8. Control de versiones

Cada vez que se realiza un cambio sobre la transcripción de una pagina, Transcriptor realiza una instantánea (snapshot) del estado actual del texto generado, persistiendo una versión de la transcripción. Esta característica le permite a la plataforma mantener, a medida que se van realizando modificaciones sobre el texto, un listado de cambios del documento que además llevan la trazabilidad de quien y cuando los realizó.

Estas versiones pueden ser consultadas por los transcriptores a modo de historial de cambios de una pagina y realizar comparaciones entre estas, mostrando las lineas agregadas, modificadas o eliminadas.

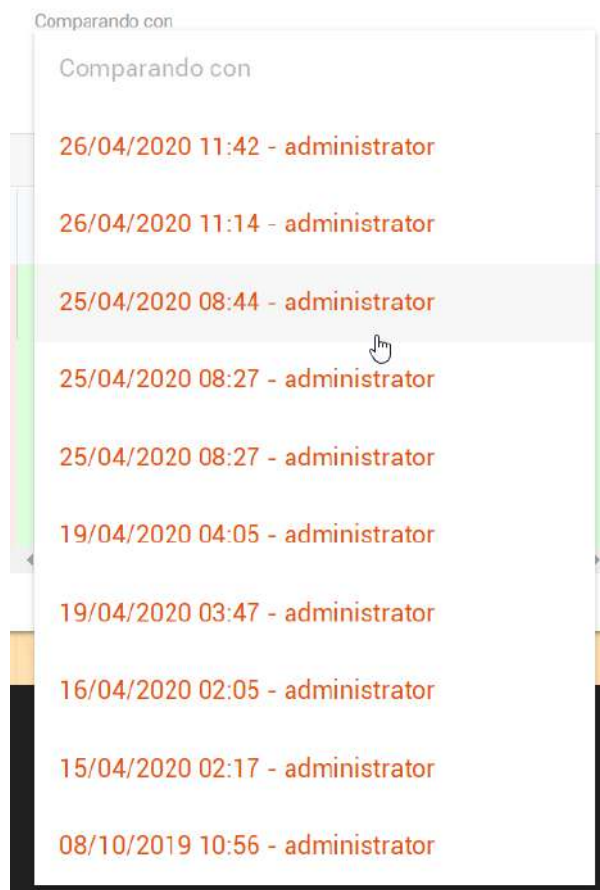


Figura 7.57: Listado de selección de versiones a comparar.


```

1  diffVersion() {
2      // Genera la cadena Diff que para realizar la comparación.
3      this.diffString = jsDiff.createPatch(this.page.title, this.pageVersion.transcription, this.page.source_text);
4
5      // Utilizando la cadena creada renderiza el componente de comparación.
6      let outputHtml = Diff2Html.getPrettyHtml(this.diffString, {
7          inputFormat: 'diff',
8          showFiles: false,
9          matching: 'words',
10         outputFormat: 'side-by-side'
11     });
12     this.outputHtml = outputHtml;
13 }

```

Figura 7.58: Método diffVersion, utilizado para generar la comparación entre versiones.

Para realizar la comparación, el usuario selecciona la versión que desea utilizar (figura 7.57) las mismas se identifican con el nombre de usuario y la fecha y hora en la cual se cargaron. Luego Transcriptor invoca al método de componente diffVersion que, haciendo uso de librería jsDiff, utiliza el texto de la versión actual y la seleccionada para generar una cadena (string) con un formato denominado diff o patch, el cual contiene una especificación de la diferencia entre ambos fuentes (fig. 7.58). A continuación y siguiendo con la ejecución del método, se hace uso de la librería diff2html que nos permite, en base a la cadena diff generada anteriormente, mostrar de una manera amigable para el usuario los cambios entre ambos textos, renderizando en paneles de comparación lado-a-lado (side-by-side) dichas diferencias (fig. 7.59).



Figura 7.59: Comparación de contenido de versiones.

De esta forma, este componente permite potenciar el trabajo colaborativo facilitando la auditoría los cambios y permitiendo un control de la actividad de los colaboradores en la transcripción de la pagina.

7.5.9. Definición de capas semánticas

Como se definió en capítulos anteriores, una capa nos ayuda a separar secciones o contextos de un manuscrito. Definiremos capas semánticas a aquellas capas que están compuestas de marcas semánticas.

Las marcas semánticas son marcas que en lugar de describir solo texto, pueden describir un objeto del mundo real, como puede ser una persona, una organización, un libro, un evento, un lugar, entre muchos otros, con la finalidad de poder representar esos elementos que no son fáciles de expresar en una transcripción.

Para generar las marcas semánticas, Transcriptor interactúa con su API Rest, que proveerá los tipos de metadatos necesarios para definir estas marcas semánticamente.

Por su flexibilidad y amplitud de conceptos definidos, para este ejemplo se utilizó la ontología schema.org, que provee un lenguaje lo suficientemente amplio como para representar las propiedades de los objetos del mundo real. Schema.org ofrece una jerarquía de tipos de datos, como podemos ver en la figura 7.60, que parte del tipo “Thing” y tiene como hijos a otras clases como “Action”, “Place”, “Person”, “CreativeWorks”, entre otros, estos a su vez tienen subclases, completando una jerarquía.

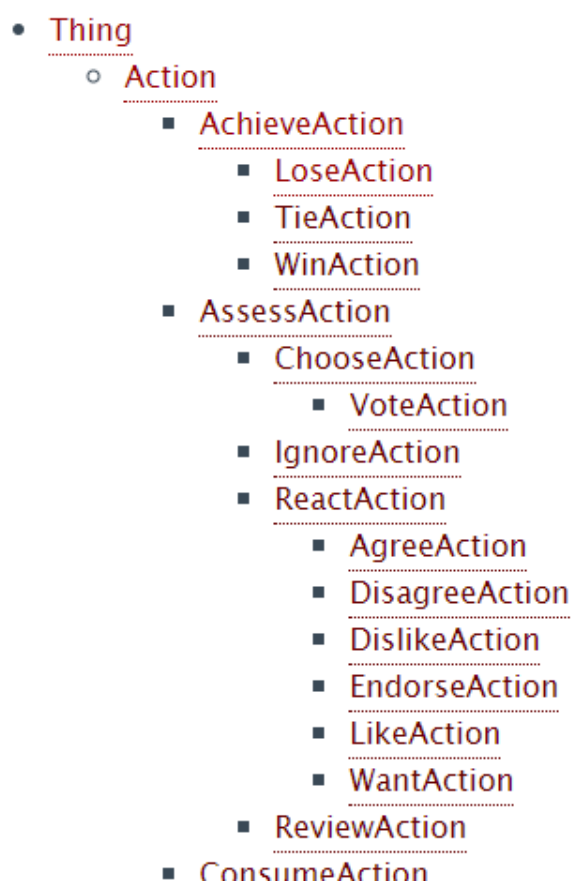


Figura 7.60: Jerarquía de clases de schema.org

A través de le servicio list_classes (descrito en la sección), Transcriptor nos provee el listado de tipos de la ontología, junto a sus cadena de jerarquía completa.

```
public getAllTypes() {
  const userAction = async () => {
    const response = await fetch(SchemeUtils.schema_tree);
    const json = await response.json();
    return json;
  };
  return userAction();
}
```

Figura 7.61: Función de obtención de tipos de schema.org

Una vez obtenido el listado de tipos, Transcriptor los presenta por niveles de la jerarquía como se muestra en la figura 7.62, permitiendo navegar y elegir el tipo más adecuado de forma intuitiva:

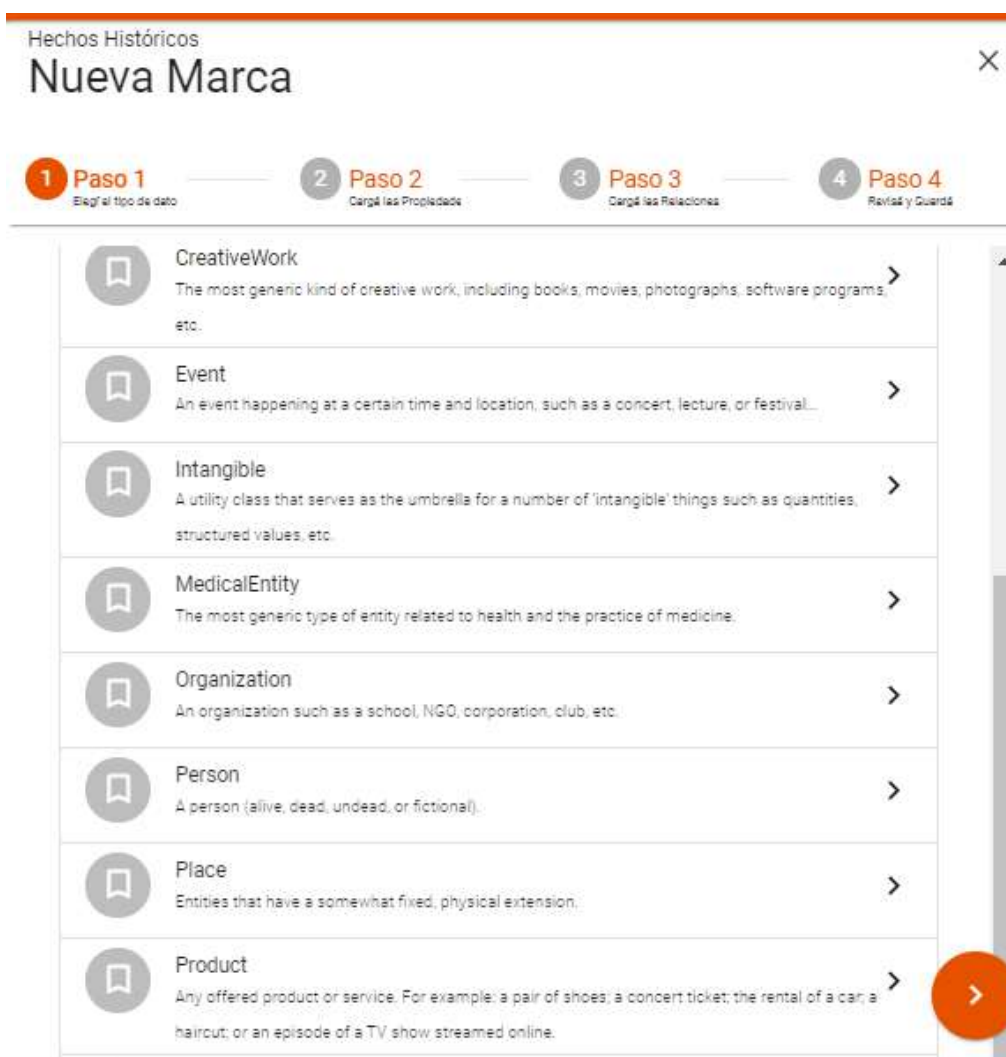


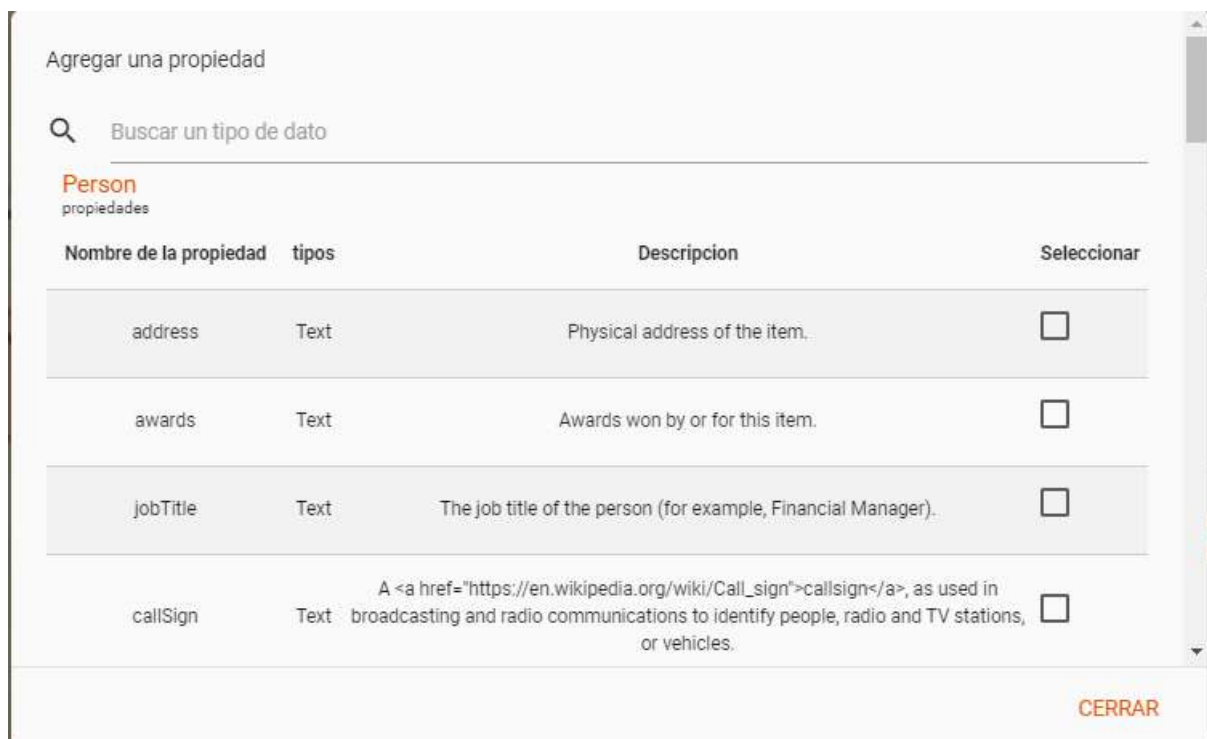
Figura 7.62: Listado de tipos cargados en transcriptor.

Cuando se ha identificado y seleccionado el tipo que se necesita, Transcriptor debe conseguir los atributos y relaciones que lo componen, esto se realiza mediante otro servicio como se muestra en la figura [7.63](#)

```
public getType(schema) {  
  const userAction = async () => {  
    const response = await fetch(SchemaUtils.schema_prefix + schema + '.jsonld');  
    const json = await response.json();  
    return json;  
  }  
  return userAction();  
}
```

Figura 7.63: Método de obtención de la descripción de un tipo puntual.

Una vez obtenidas las relaciones y atributos del tipo de schema seleccionado, Transcriptor procederá a mostrarlo como se puede ver en la figura [7.64](#), dando la oportunidad de armar un objeto con los atributos y relaciones necesarias para representar la marca en cuestión.



Agregar una propiedad

Q Buscar un tipo de dato

Person
propiedades

Nombre de la propiedad	tipos	Descripcion	Seleccionar
address	Text	Physical address of the item.	<input type="checkbox"/>
awards	Text	Awards won by or for this item.	<input type="checkbox"/>
jobTitle	Text	The job title of the person (for example, Financial Manager).	<input type="checkbox"/>
callSign	Text	A call sign , as used in broadcasting and radio communications to identify people, radio and TV stations, or vehicles.	<input type="checkbox"/>

CERRAR

Figura 7.64: Listado de propiedades para el tipo Person.

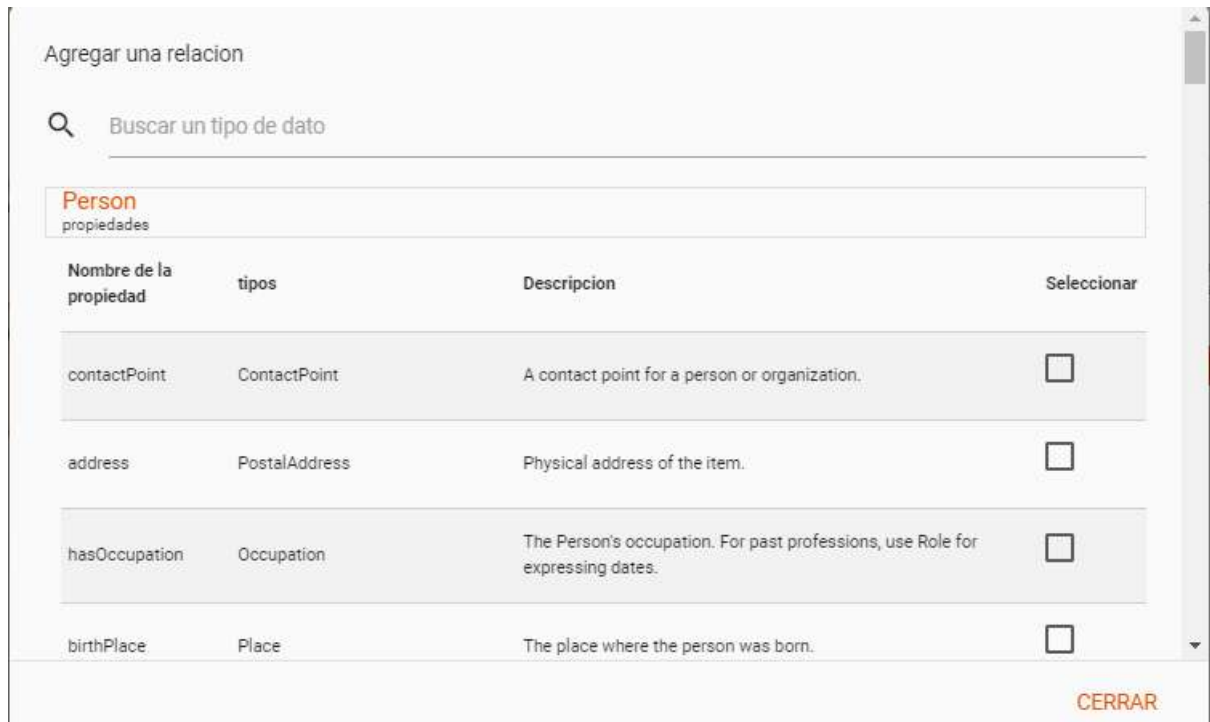


Figura 7.65: Listado de relaciones para el tipo Person.

Una vez que se carga la información en los atributos y las relaciones de la marca, se procede con el guardado, generando como resultado un objeto en formato JSONLD (como se puede ver en la figura 7.66). Se optó por utilizar JSONLD porque es un formato simple, legible y fácil de manipular, es posible además utilizar librerías de procesamiento como jsonld.js en la generación y lectura.

```
"@context": {
  "schema": "http://schema.org/",
  "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
  "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
  "transcriptor": "http://transcriptor-demo.com/"
},
"@id": "http://transcriptor-demo.com/semantic-contribution-1592275006555",
"@type": "schema:NoteDigitalDocument",
"schema:mainEntity": {
  "@id": "transcriptor:Joaquin_Turina_Perez",
  "@type": "schema:Person",
  "schema:familyName": "Turina Perez",
  "schema:givenName": "Joaquin",
  "schema:name": "Joaquin Turina Perez",
  "schema:parent": {
    "@id": "transcriptor:Joaquín_Turina_y_Areal",
    "@type": "schema:Person",
    "schema:name": "Joaquín Turina y Areal",
    "rdfs:label": "Joaquín Turina y Areal"
  },
  "rdfs:label": "Joaquin Turina Perez"
},
"rdfs:label": "Joaquin Turina Perez"
```

Figura 7.66: Definición de una instancia de la clase Person utilizando JSONLD.

7.5.10. Método de transcripción: Transcripción semántica

Para generar una transcripción semántica, Transcriptor dispone de una serie de pasos para realizarla. En primer lugar se debe crear una capa semántica para separar un conjunto de marcas del documento.

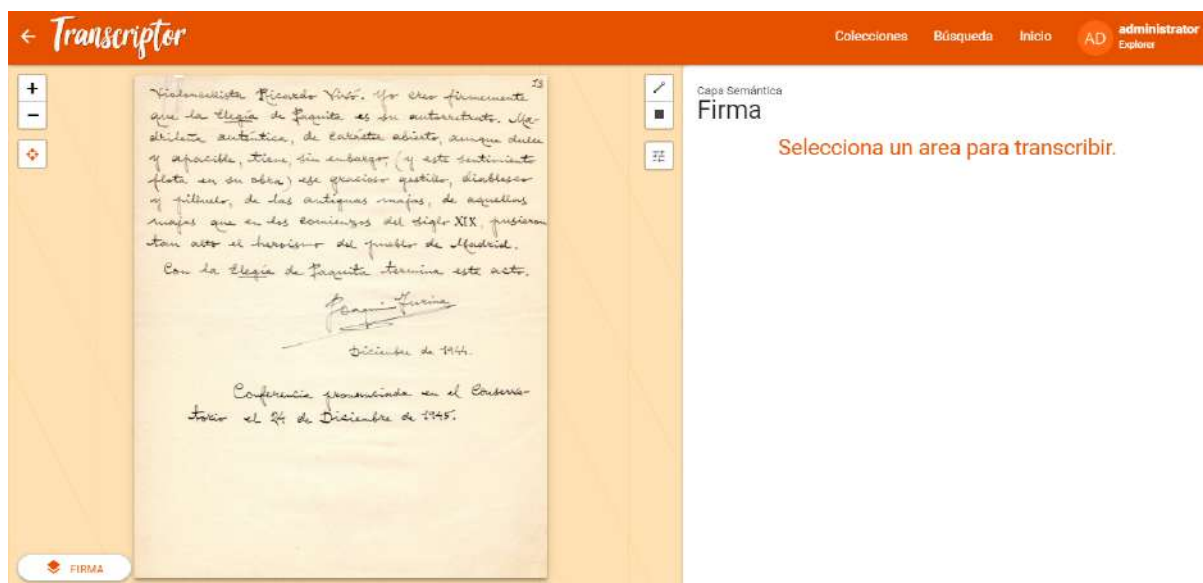


Figura 7.67: Creación de capa “Firma”

Una vez creada la capa, Transcriptor solicitará que se agregue una marca, seleccionando un área del documento en para hacer referencia a un objeto del mundo real. Algunos ejemplos pueden ser marcar donde aparezca una firma en el documento, para agregar información de una persona que hace referencia la firma o agregar información del lugar donde se hace la firma, agregando coordenadas geográficas.

Una vez seleccionada el área a transcribir, se podrá asociar contenido de dos maneras: creando nuevo contenido en transcriptor o buscar contenido previamente guardado.



Figura 7.68: Selección y asociación de entidad precargada a la marca.

Para crear nuevo contenido, se presenta una serie de cuatro pasos para generar un objeto que hace referencia algo del mundo real, en este caso la persona puede estar asociada a la firma. para el primer paso, Transcriptor listará una jerarquía de tipos seleccionables

para representar la marca



Figura 7.69: Ejemplo de marcado de una firma.



Figura 7.70: Selección de tipo para representar el contenido de la marca.

Como se puede ver en la figura [7.70](#), como primer paso, tendremos que elegir el elemento que queremos representar en la marca, uno de los tipos que se aproxima a este es Person, el cual representa una persona. Una vez seleccionado el tipo, podremos se presentará el siguiente paso, automáticamente transcriptor realizará la carga de los atributos que pueden ser asignados al tipo seleccionado.



Figura 7.71: Carga de propiedades de la nueva entidad.

Por defecto y como campo obligatorio, es necesario asignar un nombre a la entidad, ya que sirve para identificarla de una manera más legible por humanos (human readable), a continuación se puede construir a medida el objeto, agregando o quitando propiedades de tal manera que se acerque a una representación mas real del objeto en cuestión. En esta etapa cada propiedad es de tipo básico, esto es, que solo puede tener asignado un rango de valores literal, como pueden ser: texto, booleano, numérico, fecha y hora.

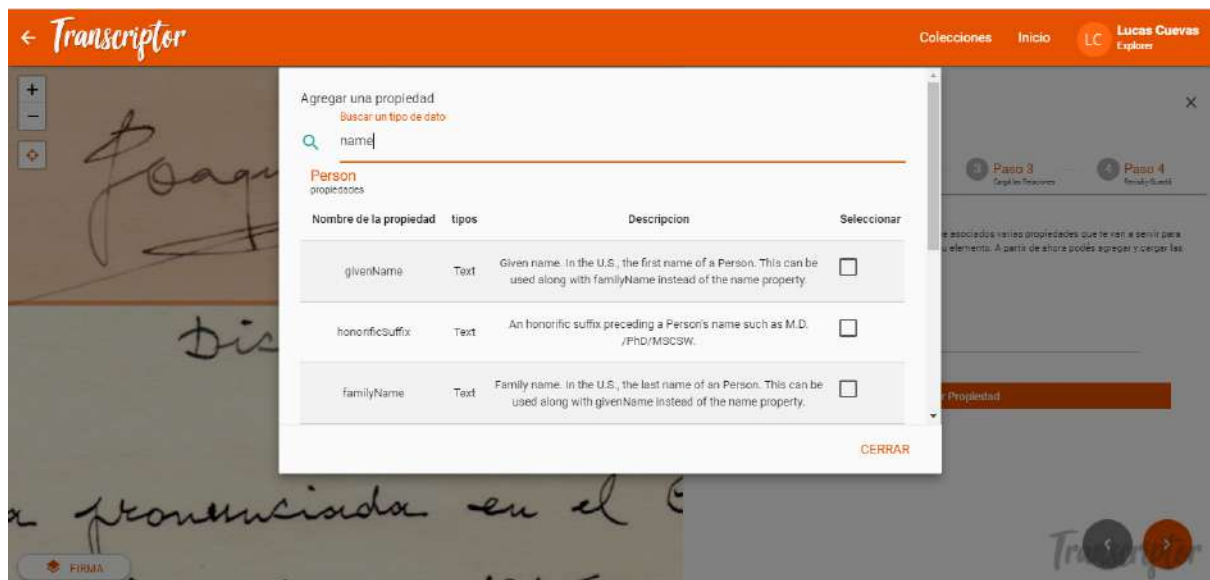


Figura 7.72: Ejemplo de búsqueda de propiedades para el tipo Person

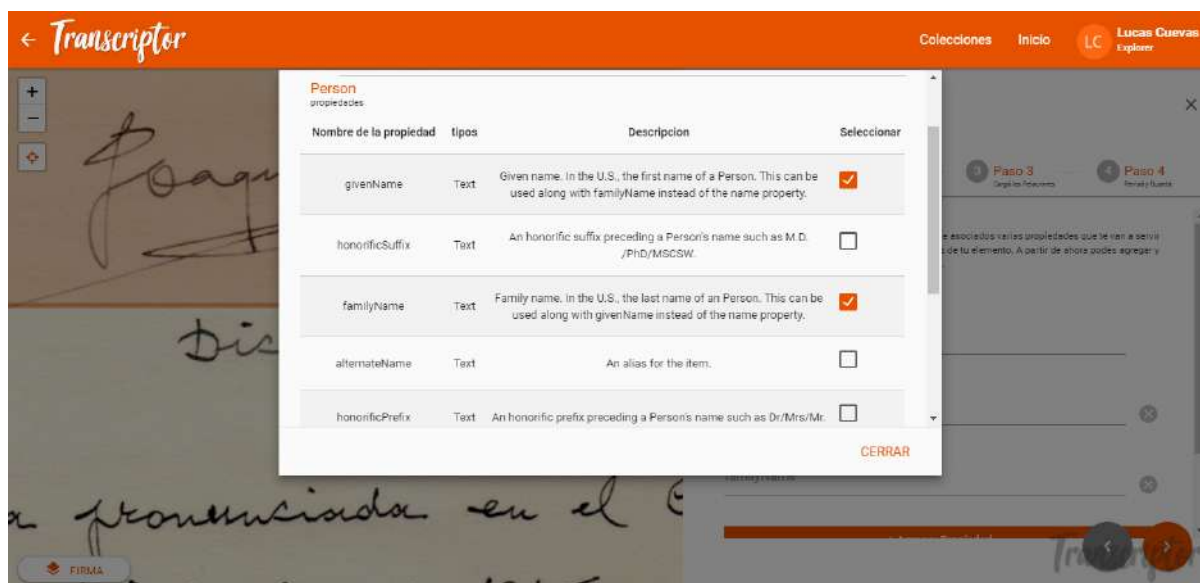


Figura 7.73: Ejemplo de selección de propiedades para el tipo Person

Una vez seleccionadas las propiedades básicas que necesitamos para representar al tipo, es necesario cargar las propiedades con datos, como se muestra en la figura [7.73](#), según corresponda su tipo.

Completado el paso de carga de datos, se habilita a continuar al siguiente paso: la carga de relaciones.

La carga de relaciones consiste en asociar otras entidades al objeto principal. Transcriptor brinda dos posibilidades de asignar relaciones, la primera de ellas es buscar contenido previamente guardado y asignarlo, la otra manera es crear el contenido a través de la relación. Para este primer caso nos centraremos en la carga de nuevo contenido, por ejemplo, para un tipo Person, se puede crear la relación con padres, lugar de nacimiento, una profesión, premios etc.

Una vez seleccionadas las relaciones para el tipo elegido (ver figura [7.74](#)), se debe seleccionar de qué manera se le asigna la relación al objeto (ver figura [7.75](#)), ingresando un texto se puede buscar contenido guardado para relacionar, la otra opción es crear una entidad nueva, para esto hay que crear una entidad nueva y luego seleccionar las propiedades que puede tener esa relación (ver figura [7.76](#)) esta última es similar al paso previo, pero en lugar de cargar propiedades básicas en el objeto que queremos representar, se cargaran las propiedades en la relación que le queremos asignar al objeto en cuestión.

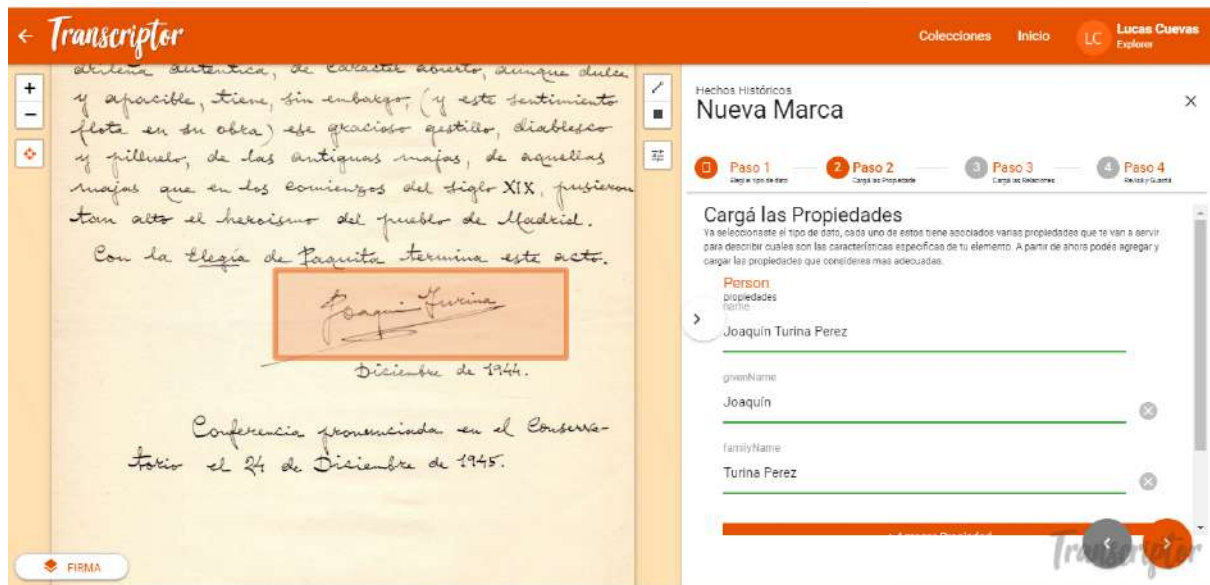


Figura 7.74: Ejemplo de proceso de carga de información para nueva entidad de tipo Person.

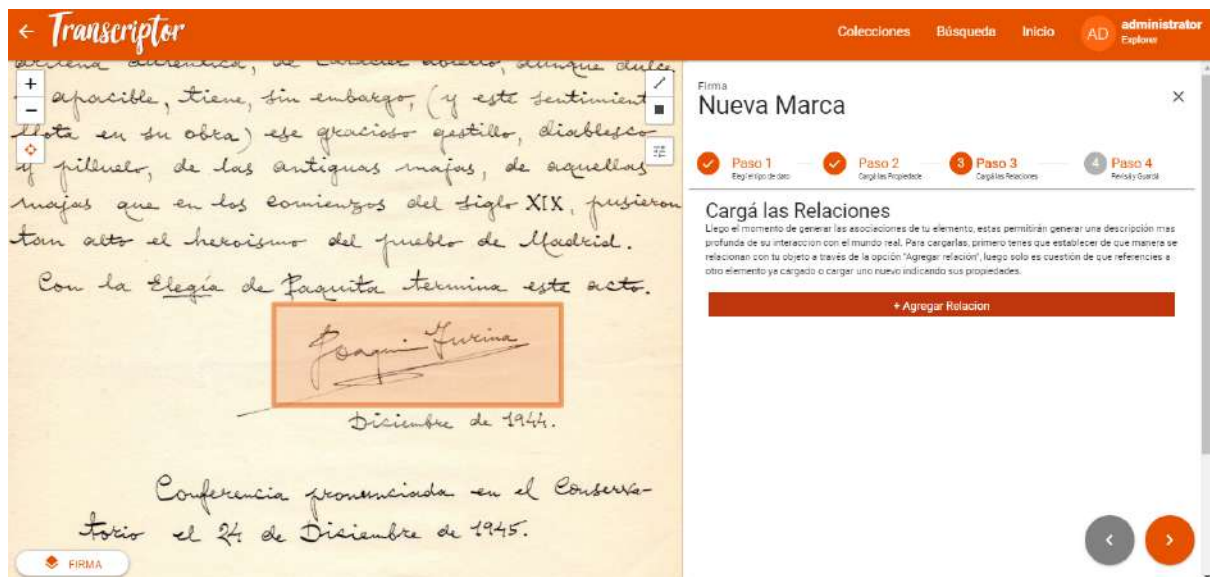


Figura 7.75: Proceso de carga de relaciones.

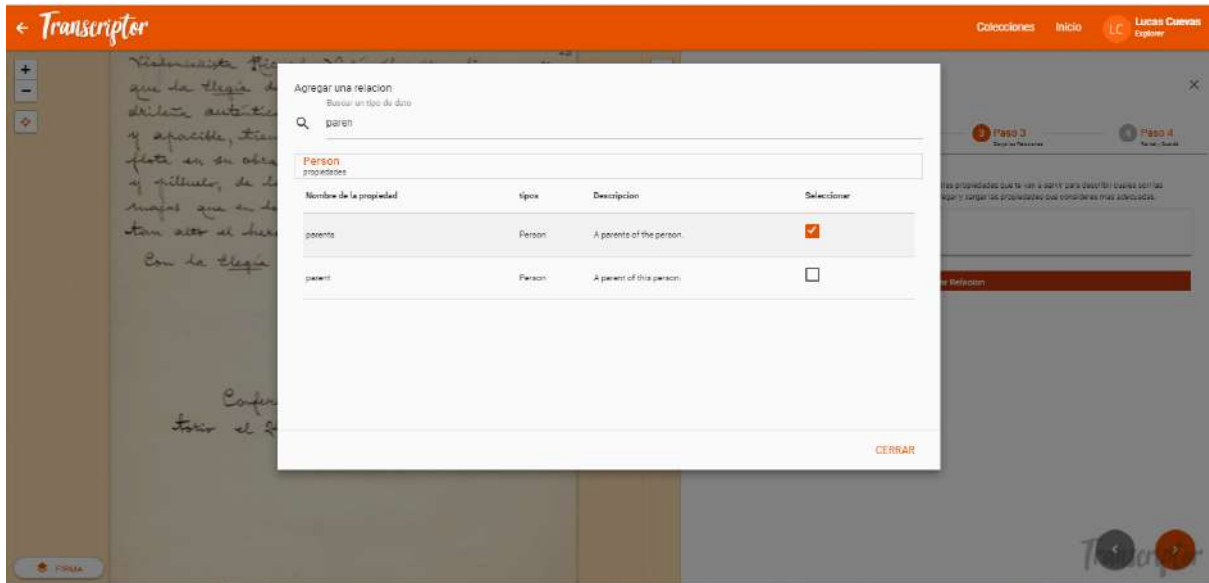


Figura 7.76: Selección de atributos de relación.



Figura 7.77: Asignación de relación a la nueva entidad.

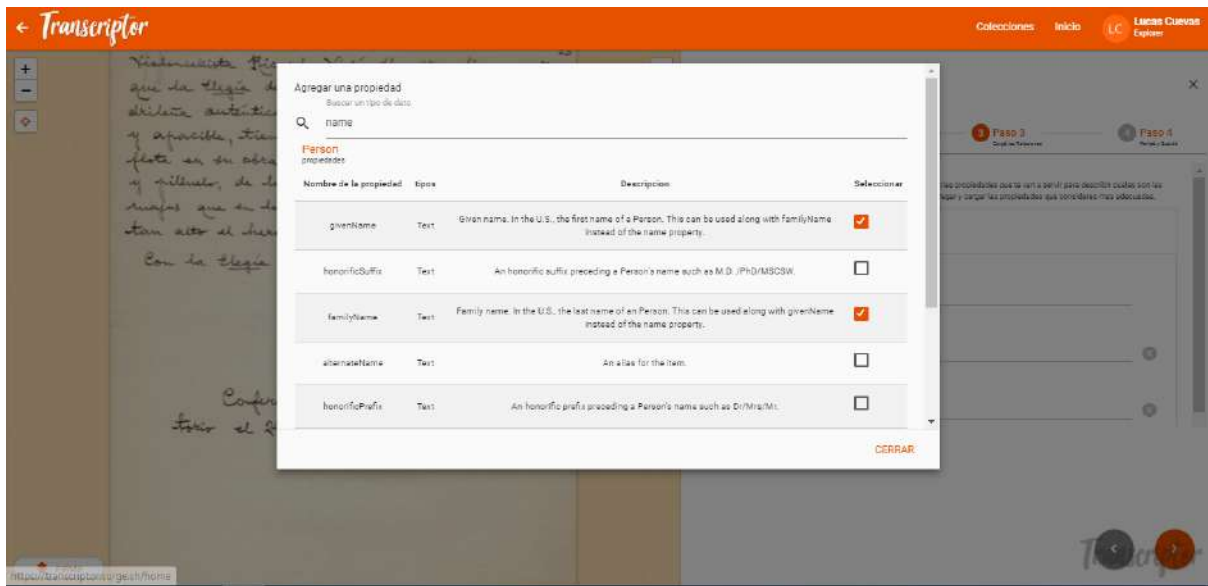


Figura 7.78: Carga de propiedades de una nueva relación embebida.

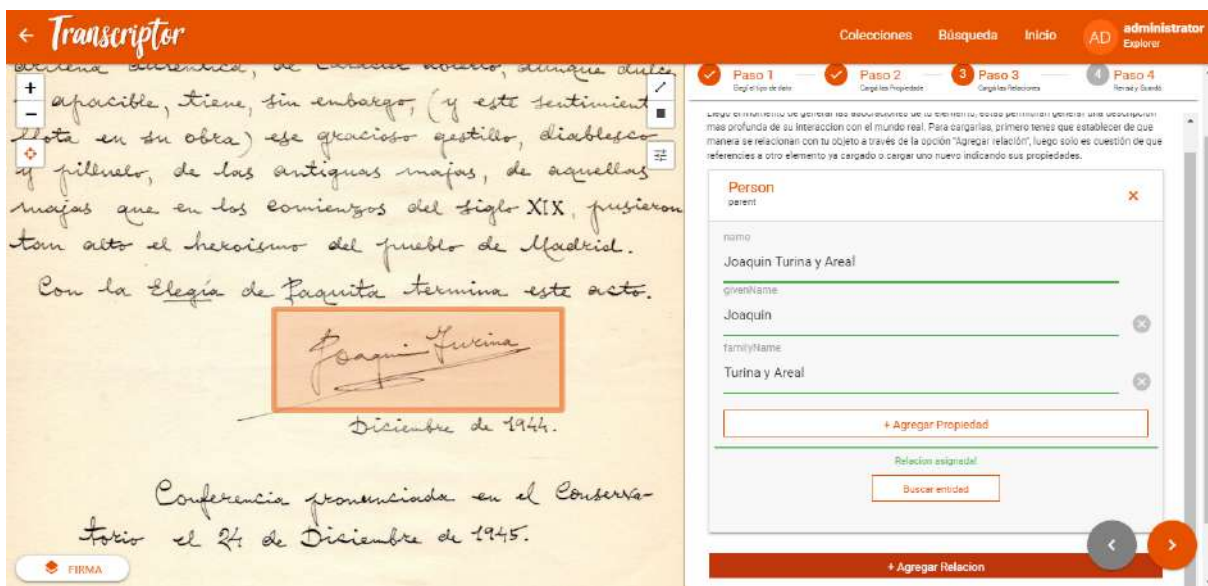


Figura 7.79: Carga de datos en la relación.

Una vez generada la relación y cargada su información, esta se guarda con los datos cargados y se asigna al objeto principal. Ya se puede avanzar al siguiente y último paso: la vista previa y el guardado de la marca como se puede representar en la siguiente figura.

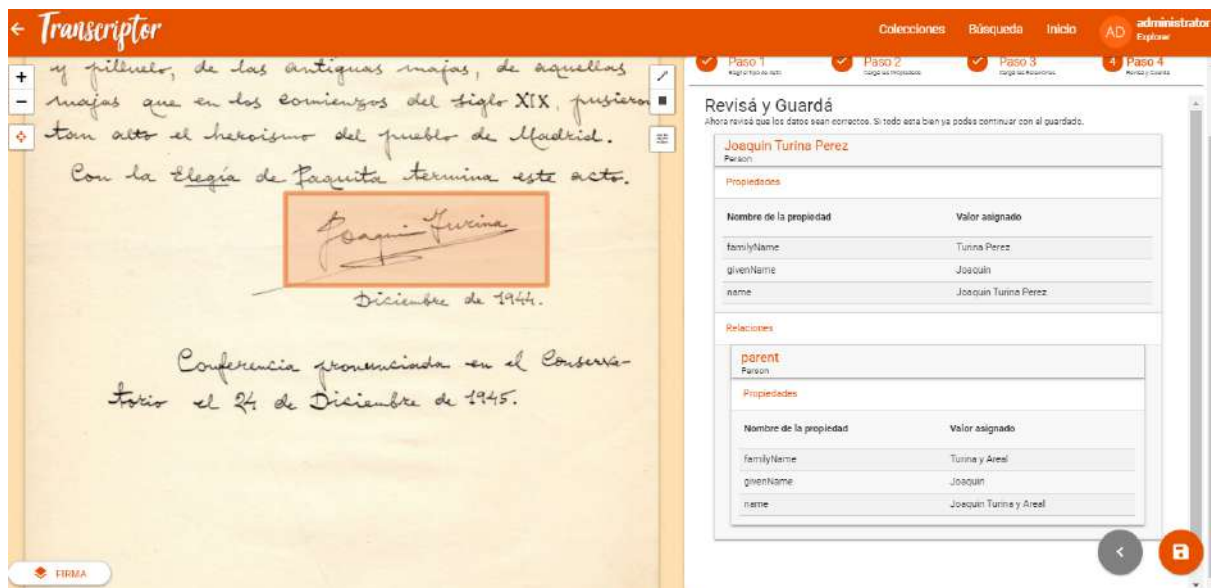


Figura 7.80: Vista previa a guardar la nueva entidad.

En este paso, podremos corroborar que no nos haya faltado nada por describir, en tal caso, es posible volver sobre los pasos anteriores para agregar o quitar información. De no haber inconvenientes, se guarda la marca creada, finalizando el proceso de transcripción semántica. Una vez guardada, se podrá acceder al detalle de la marca con contenido semántico desde la vista principal de la capa.

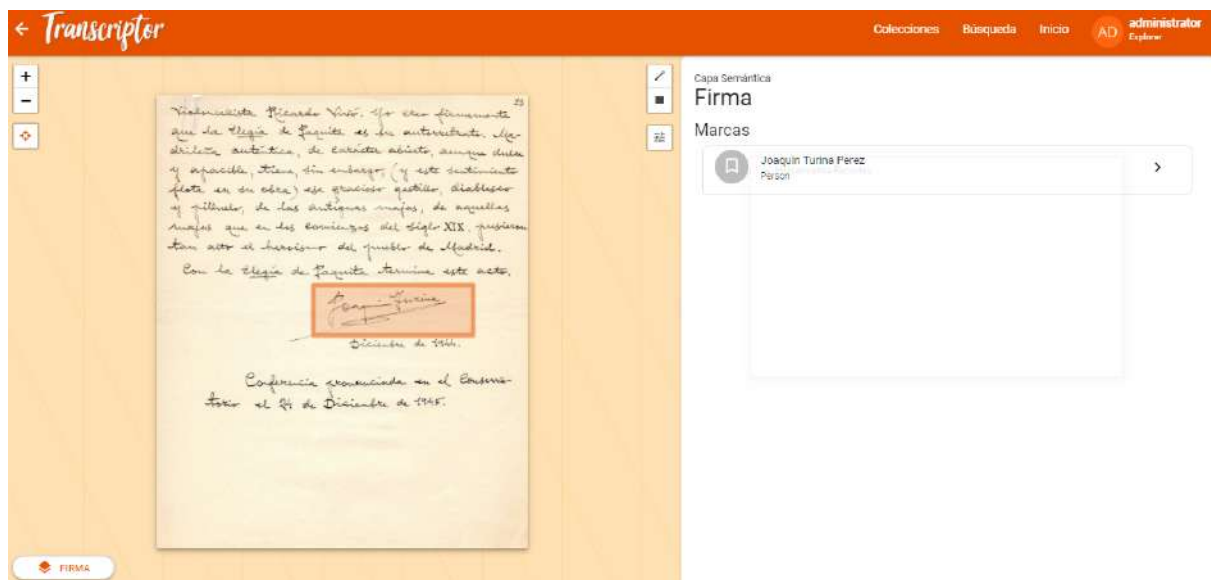


Figura 7.81: Listado de marcas de marcas semánticas la capa del documento.



Figura 7.82: Detalle del contenido semántico de una marca ya guardada.

Hasta aquí se detallo el flujo de carga desde cero de una entidad en Transcriptor pero, como se mencionó al principio de esta sección, también podemos asignar a una marca una entidad ya guardada, por ejemplo, tomando el siguiente documento que presenta la siguiente firma:

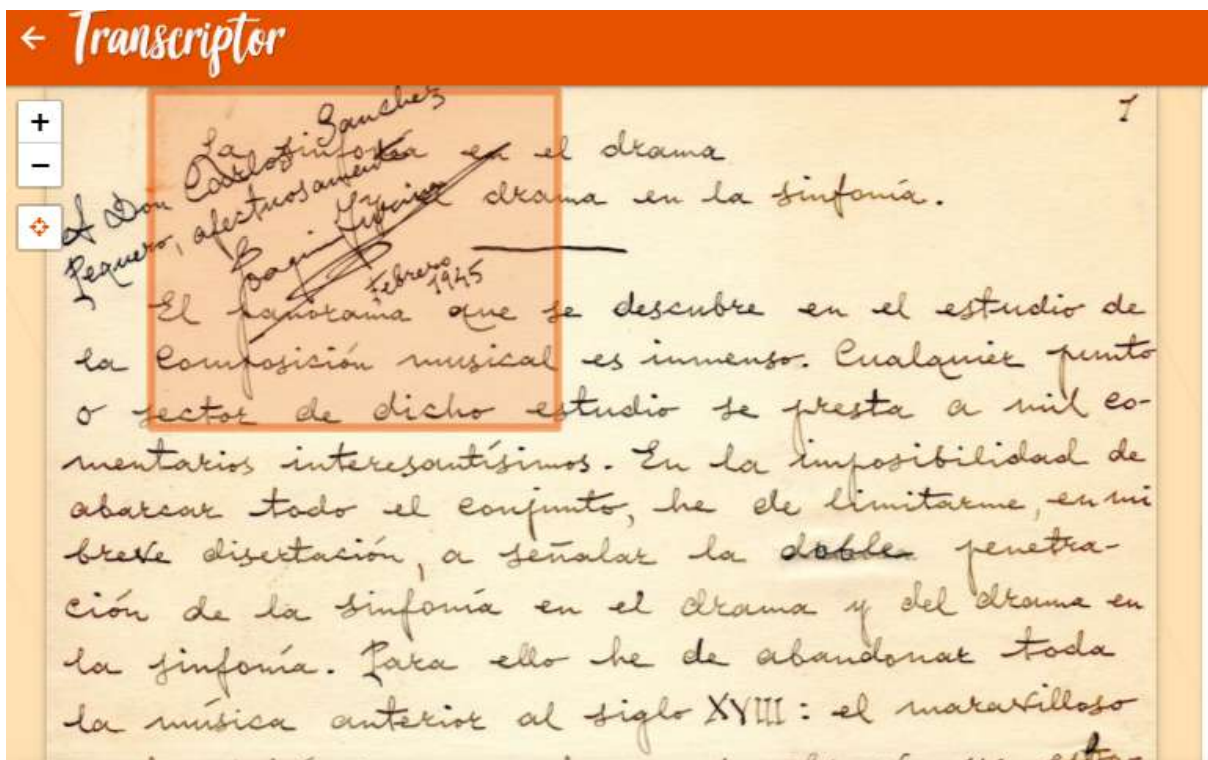


Figura 7.83: Ejemplo de firma a relacionar con una entidad pre-guardada.

La firma hace referencia a una entidad de tipo persona que previamente se guardó en Transcriptor y por lo tanto es posible relacionar a la firma. De esta manera, una vez creada

la capa y seleccionando el área de la firma volvemos al punto de partida de la creación de la marca y para este caso se realizará la búsqueda de la entidad persistida.

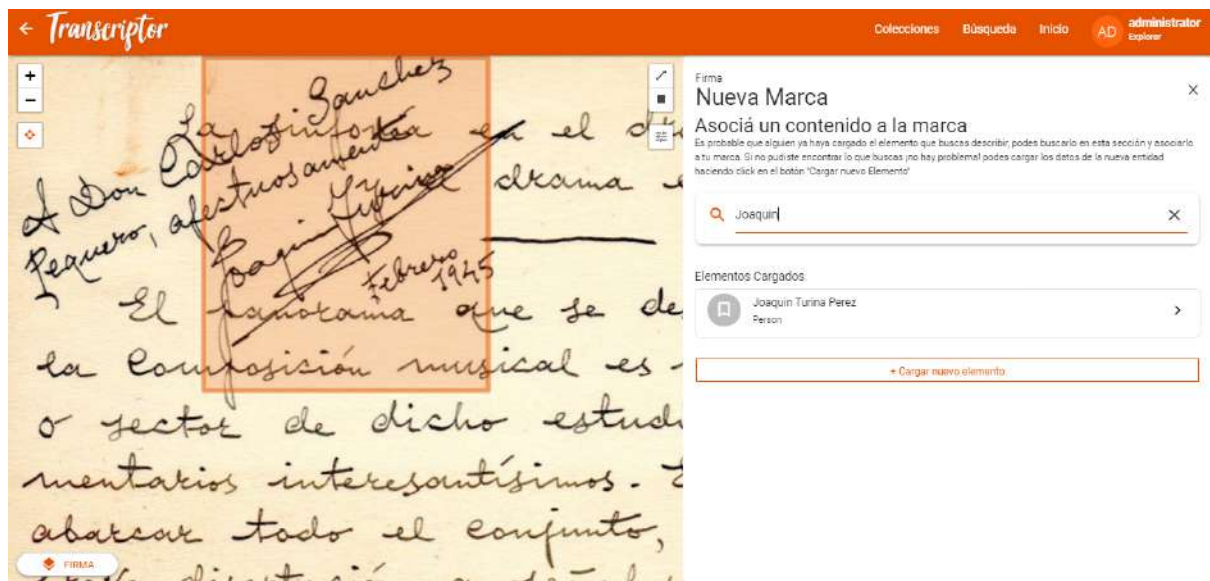


Figura 7.84: Búsqueda y selección de una entidad previamente guardada.

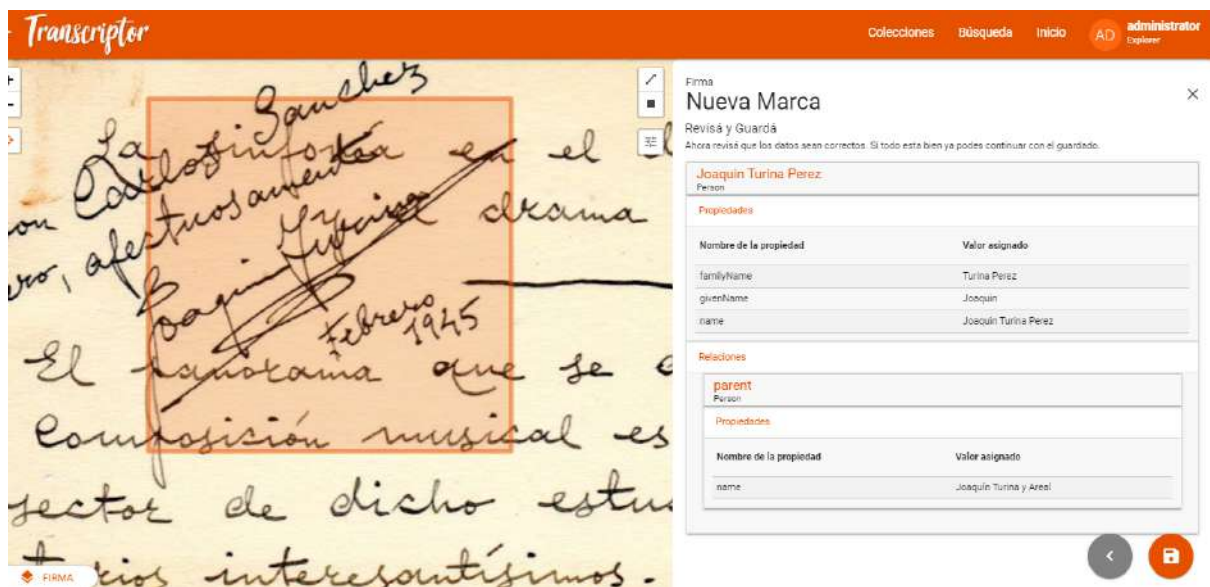


Figura 7.85: Vista previa al guardado de un objeto desde la búsqueda.

Cuando se encuentran coincidencias con el objeto, es posible visualizar el detalle del mismo (véase figura 7.84). Una vez que se corrobora que sea la entidad a relacionar, se procede a guardar Finalizado el proceso de asignación de marca por búsqueda (véase figura 7.85).

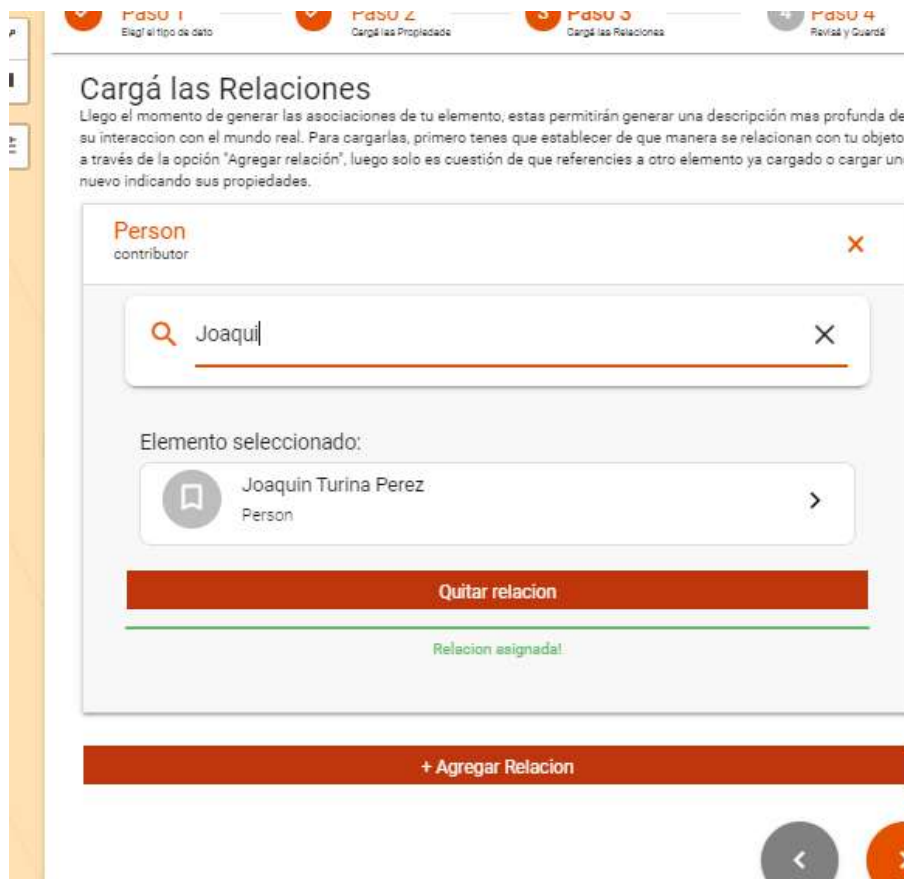


Figura 7.86: Búsqueda y asignación de un objeto existente en una relación.

La búsqueda de un objeto ya existente en transcriptor también está presente en la asignación de relaciones, cuando se crea una entidad esta búsqueda se realiza sobre el tipo que corresponde a la relación.

Una vez seleccionado el objeto se podrá ver el detalle y guardar la relación, teniendo la posibilidad de corregir si no es el objeto buscado.

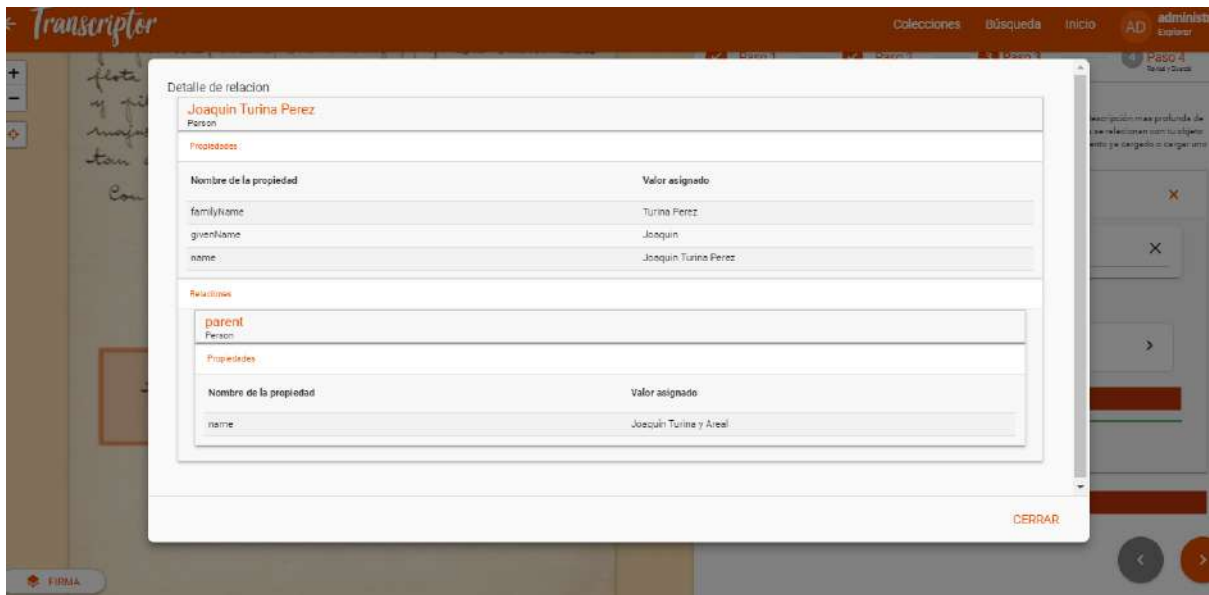


Figura 7.87: Detalle del objeto encontrado.

7.5.11. Internacionalización en Cliente (Client-Side i18n)

De la misma manera que en Transcriptor-Backend, Transcriptor-Frontend también implementa mecanismos de internacionalización en las vistas y componentes que lo integran.

Para implementar este mecanismo se utilizó el módulo ngx-translate para Angular, el cual brinda la posibilidad de definir archivos de localización y referenciarlos de manera programática.

Los archivos de localización (como se explicó anteriormente en el apartado de internacionalización de Transcriptor-Backend) son archivos en los cuales se definen todos los textos de un sitio para un idioma o dialecto en particular, están escritos en formato clave-valor, donde el valor es el texto a mostrar y la clave es el identificador para poder referenciar al texto especificado, en el caso de Transcriptor-Frontend los archivos se representaron en notación JSON.

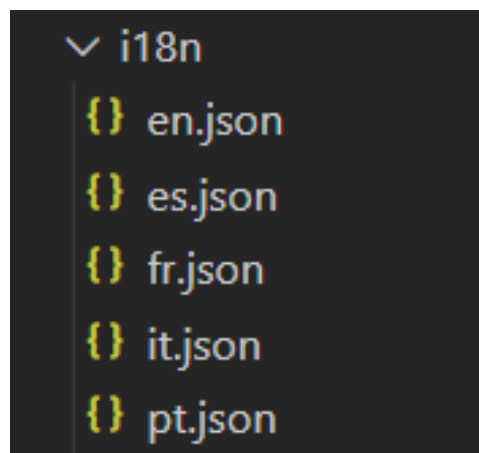


Figura 7.88: Archivos de localización, uno por cada idioma soportado en el proyecto Transcriptor-Frontend.

```

1 // es.json
2
3 {
4   "app":{
5     "button":{
6       "close":"Cerrar",
7       "cancel":"Cancelar",
8       "save":"Guardar",
9       "saveChanges":"Guardar Cambios",
10      "delete":"Eliminar",
11      "show":"Ver"
12    }
13  },
14  "home":{
15    "banner":"\`Transcriba y Traduzca Manuscritos de manera práctica, colaborando en línea con muchos otros usuarios\`",
16    "image":{
17      "left":{
18        "title":"Trancribir",
19        "text":"Debido a que Transcripator se ejecuta en su navegador, puede leer y transcribir materiales desde cualquier lugar."
20      }
21    }
22  }
23 }

```

```

1 // en.json
2
3 {
4   "app":{
5     "button":{
6       "close":"close",
7       "cancel":"cancel",
8       "save":"save",
9       "saveChanges":"save changes",
10      "delete":"delete",
11      "show":"show"
12    }
13  },
14  "home":{
15    "banner":"\`Transcribe and Translate manuscripts comfortably and online collaborating with many other users\`",
16    "image":{
17      "left":{
18        "title":"Transcribe",
19        "text":"because Transcripator runs in your browser, you can read and transcribe materials from anywhere."
20      }
21    }
22  }
23 }

```

Figura 7.89: Fragmentos de los archivos de localización en español e inglés.

La internacionalización se realiza en los distintos componentes visuales de la plataforma y, para poder utilizarla es necesario solicitar la “traducción” referenciando a la clave desde la vista de los mismos, para esto la librería `i18n` utilizada nos provee del filtro Angular `translate`, el cual nos permite obtener dicho valor. En la figura [7.90](#) se incluye un ejemplo de este procedimiento.

```

1 <div class="purpose">
2   <div class="row">
3     <div class="col 14 offset-14">
4       <h5 class="flow-text">{{'home.banner' | translate}}</h5>
5     </div>
6   </div>
7 </div>

```

Figura 7.90: Ejemplo extraído de la vista inicial del frontend, mostrando la utilización del filtro `translate`.

Finalmente, para poder establecer en qué idioma mostrar el sitio, durante la carga se ejecuta un proceso que inicializa las preferencias de internacionalización. Para esto, la librería `i18n` provee un método que permite detectar cual es el idioma utilizado en el

navegador (línea 3 en figura 7.91) consultando la configuración del mismo. Luego, una vez obtenido el acrónimo del lenguaje configurado, se usa este valor para seleccionar el archivo de localización a utilizar durante la resolución de las referencias en las vistas y, de esta forma, mostrar el idioma según las preferencias de cada usuario.

```
1 constructor(public translate: TranslateService) {
2     // configura el idioma por defecto
3     translate.setDefaultLang('en');
4     // obtiene el idioma configurado en el navegador
5     const browserLang = translate.getBrowserLang();
6     // configura la librería para utilizar el idioma detectado
7     translate.use(browserLang);
8 }
```

Figura 7.91: Inicialización de las preferencias de internacionalización.

De esta manera, al renderizar la vista se resuelven las referencias definidas con el archivo de idiomas correspondiente y se presenta con el idioma configurado según el navegador.

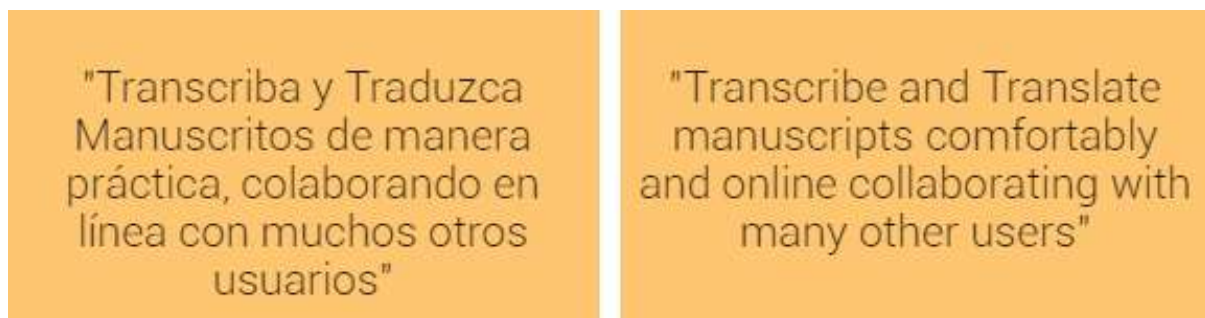


Figura 7.92: Fragmento de la página inicial, en idioma español o inglés según el idioma del navegador.

7.5.12. Despliegue e Instalación

Para utilizar Transcriptor-Frontend es necesario tener, como primera instancia una versión instalada y operativa de Transcriptor-Backend a la cual la aplicación realizará las peticiones necesarias para su funcionamiento. Luego, es necesario instalar y configurar la aplicación para poder utilizarla en ambientes de desarrollo y/o productivos.

Instalando Dependencias

En su implementación, Transcriptor-Web hace uso de varias librerías y módulos que definen la estructura del proyecto y facilitan la realización de tareas, entre las que tenemos ejemplos como el framework Angular o librerías de uso puntual como JSDiff, los cuales su desarrollo está ligado a terceros y se consideran como dependencias (o packages) para que el proyecto pueda cumplir su función.

De esta manera, para mantener un correcto orden y establecer una estructura por la cual estas dependencias son requeridas, se utiliza el manejador de paquetes npm (Node

Package Manager) el cual además de gestionar la adición de nuevos paquetes al proyecto también nos provee un entorno de ejecución Javascript a través del cual podemos ejecutar procesos y tareas útiles para el proyecto.

Las dependencias utilizadas en el proyecto se mantienen definidas en un archivo llamado `package.json`, donde se encuentra toda la configuración de las versiones a instalar y a que entorno de ejecución pertenecen (producción o desarrollo).

Finalmente para realizar la instalación de las dependencias es cuestión de ejecutar el comando `npm install`, disparando la ejecución del proceso de descarga y posterior instalación de cada paquete.

Configurando el ambiente

Una vez instaladas las dependencias, es necesario realizar la configuración del ambiente, esto es, en que dominio (URL/host) se encuentra alojado Transcriptor-Backend, para configurarlo como URL base de los llamados a la API.

Esta configuración esta definida en los archivos de configuración de ambiente los cuales permiten la definición de configuraciones según el ambiente en ejecución. Transcriptor utiliza dos archivos, uno para entornos de desarrollo y otro para entornos productivos (respectivamente `environment.ts` y `environment.prod.ts`).

```
1 export const environment = {
2   production: false,
3   apiUrl: 'http://localhost:3000', // URL de Transcriptor-Backend
4   usePusher: false
5 };
```

Figura 7.93: Contenido del archivo `environment.ts`

Publicar en servidor

Para realizar la publicación de la aplicación en un ambiente, ya sea para desarrollo o producción, y al tratarse de un proyecto Angular, el cual se basa en el lenguaje TypeScript, es necesario compilar su código fuente para obtener los archivos finales en formato Javascript y los cuales serán servidos por el servidor al usuario. Compilar el proyecto inicia un proceso a través del entorno de ejecución de npm que minifica y compila el código en cuestión.

En el caso de ambientes locales de desarrollo, solo con ejecutar el comando `ng serve` se ejecuta la compilación y la publicación en un pequeño servidor de desarrollo incluido en Angular, permitiendo de manera rápida iniciar la app y desarrollar sobre la misma (figura 7.94).

Para ambientes de desarrollo o producción remotos, se invoca a la tarea de compilación de manera independiente, para lo que se utiliza el comando `ng build` al que, además, se le puede especificar si se esta construyendo una compilación para entornos de producción o desarrollo a través del flag `env`. El código compilado por defecto se guarda en un directorio dentro del proyecto.

Finalizado el proceso, los archivos resultantes ya se encuentran listos para publicarse en cualquier servidor de contenido estático, como por ejemplo: Apache o Nginx.

```
D:\transcriptor-web (dashboad-ui -> origin)
λ ng serve

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Date: 2020-04-30T08:20:33.037Z
Hash: c151d6c9d0e5e456f664
Time: 38320ms
chunk {main} main.js, main.js.map (main) 671 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 240 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.22 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 350 kB [rendered]
chunk {styles} styles.js, styles.js.map (styles) 634 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 9.74 MB [initial] [rendered]
i | wdm | Compiled successfully.
```

Figura 7.94: Salida del comando `ng serve`, donde se puede observar el inicio del servidor de desarrollo y el proceso de compilación (`ng build`).

Capítulo 8

Evaluaciones y Presentaciones

En este capítulo se abordará el proceso de construcción y realización de pruebas sobre la plataforma en distintos ámbitos y utilizando métodos de evaluación tanto empíricos como técnicos. También se incluyen las presentaciones en congresos y las demostraciones públicas del uso del sistema.

8.1. Evaluación

8.1.1. Objetivo

Al tratarse de una aplicación abierta a la colaboración del público en general, la usabilidad juega un papel muy importante en la experiencia del usuario al utilizar la plataforma. La facilidad de encontrar herramientas o la simplicidad con la que se aborda la realización de una tarea compleja, pueden ser definitivos para que una persona se sienta motivada a participar de las actividades que se proponen. En este contexto se busca evaluar la plataforma en términos de usabilidad y experiencia del usuario (UX).

8.1.2. Evaluación de usabilidad con usuarios

Objetivo

Esta prueba busca obtener información acerca del nivel de facilidad de uso de la plataforma para evaluar la capacidad que ésta tiene para cumplir su propósito. Para esto, se propusieron una serie de actividades las cuales están enfocadas en el éxito/fracaso del usuario para realizarlas y cuánto tiempo le conlleva ejecutarla. Para ello se buscará responder las siguientes preguntas:

1. ¿La plataforma es simple de usar?
2. ¿Puede resultar complejo aprender a usar la plataforma?
3. ¿Se podría considerar a Transcriptor como una solución innovadora en el campo de la transcripción?
4. ¿El sitio y las actividades motivan al usuario a seguir participando?

Finalmente se le presentó al participante un conjunto de preguntas que tienen como objetivo capturar la experiencia y el nivel de conformidad como usuario frente a las tareas y al entorno de trabajo del sistema.

Diseño de la prueba

En esta instancia se definió que datos se requerían recolectar, para luego analizar y sacar conclusiones. A criterio propio:

- Datos medibles a través de alguna métrica: Emplear algún mecanismo que permita medir la experiencia basándose en escalas de estudios anteriores para determinar cual es el grado de cumplimiento tiene Transcriptor en comparación con otras plataformas y si el mismo es aceptable. Existen varios métodos para obtener estos datos, entre los que se analizaron UEQ (User Experience Questionarie), SUS (System Usability Scale), SUMI (Software Usability Measurement Inventory), SUPR-Q (Standardized User Experience Percentile Rank Questionnaire) y SUM (Single Usability Metric)
- Evaluaciones y pensamientos individuales de los usuarios: También es de interés contar con datos empíricos de la experiencia de los participantes. Una manera de recabar este tipo de información es utilizar preguntas abiertas que pueden llevar al usuario a desarrollar su punto de vista del sistema y elaborar problemáticas que no son posibles de expresar a través de puntajes y escalas.

En este contexto se desarrolló una evaluación la cual esta dividida en tres partes:

- Evaluación por Tarea: un test UEQ reducido por actividad realizada por el usuario, conformado por un cuestionario de cinco preguntas, las cuales consisten en pares de adjetivos opuestos junto a niveles que marcan una escala, donde quien completa el cuestionario deberá seleccionar qué escala representa mejor su experiencia con la plataforma, cada pregunta hace referencia a propiedades de transparencia, atracción y novedad.
- Panorama general de la plataforma: Preguntas abiertas de desarrollo corto sobre las funcionalidades probadas (5 preguntas).
- Conclusiones: Test SUS completo en su versión positiva [22], que se conforma de diez ítems, donde se presentan afirmaciones que tendrán la escala de 1 a 5. El participante deberá indicar que tan de acuerdo se siente con dicha afirmación, siendo 1 completamente en desacuerdo y 5 totalmente de acuerdo.

8.1.3. Resultados esperados

Para responder las preguntas enumeradas en la sección 8.1.2 se contemplaron los siguientes expectativas de resultados:

- Pregunta 1 (¿La plataforma es simple de usar?) y Pregunta 2 (¿Puede resultar complejo aprender a usar la plataforma?).
 - Utilizaremos el puntaje de SUS comparándolo con la escala definida por Bangor, Kortum, y Miller (2008) [23] la cual se explica con mayor detenimiento en la sección “Resultados obtenidos”. Dicha escala especifica que, con un puntaje SUS entre 0 y 20.3, el nivel de usabilidad sería el “peor imaginable”, entre 20.3 y 37.7 sería “malo”, entre 37.7 y 50.9 es “regular”, entre 50.9 y 71.4 es “aceptable”, entre 71.4 y 81.5 “bueno”, entre 81.5 y 90.1 “excelente” y entre 90.1 y 100 la usabilidad sería la “mejor imaginable”. Para cumplir con las preguntas planteadas se espera que el puntaje SUS se encuentre por sobre los 70 puntos,

con lo que alcanzaría el nivel “aceptable”.

Además para estos puntos utilizaremos los coeficientes UEQ de “transparencia” que nos permiten determinar que tan simple es la plataforma para comunicar y realizar los objetivos que plantea, si el promedio de los mismos se encuentran por sobre los 3 puntos se podría considerar como que el sistema es transparente y confirmar las preguntas inicialmente presentadas.

- Pregunta 3 (¿Se podría considerar a Transcriptor como una solución innovadora en el campo de la transcripción?).
 - Para confirmar este interrogante se usará el indicador UEQ de “novedad”, el cual nos denotará el nivel de innovación que presentan las funcionalidades del sistema. Comparando el promedio calculado entre las características principales del sistema se espera que esta escala supere los 3 puntos.
- Pregunta 4 (¿El sitio y las actividades motivan al usuario a seguir participando?).
 - De la misma manera que para la pregunta 3 se tomó el indicador de “novedad” para establecer el cumplimiento, para este planteo se utilizará la escala UEQ de “atracción”. Este promedio nos permite establecer el nivel de interés que las funcionalidades del sistema despiertan en los usuarios. Obtener un puntaje promedio mayor a 3 entre todas las características permitiría corroborar este punto.

Desarrollo de la prueba

Se lanzó una convocatoria abierta para usuarios interesados en probar la plataforma. La propuesta fue realizar entrevistas virtuales individuales de una hora de duración las cuales fueron grabadas para mantener un registro de las reacciones de los participantes durante la evaluación. Las entrevistas siguieron la modalidad de explicar-hacer, de manera que se daba un espacio para explicarle la actividad y a continuación realizar la misma. Luego de cada entrevista se les envió el cuestionario desarrollado por mail el cual completaban de manera anónima a través Google Forms. Se realizaron 19 entrevistas en las que participaron personas de diversas edades, niveles de estudio y profesionales, entre los que podemos destacar especialistas informáticos, historiadores, filólogos e investigadores del área de letras (ver figura [3.1](#)).

Distribución de especialidades de los entrevistados

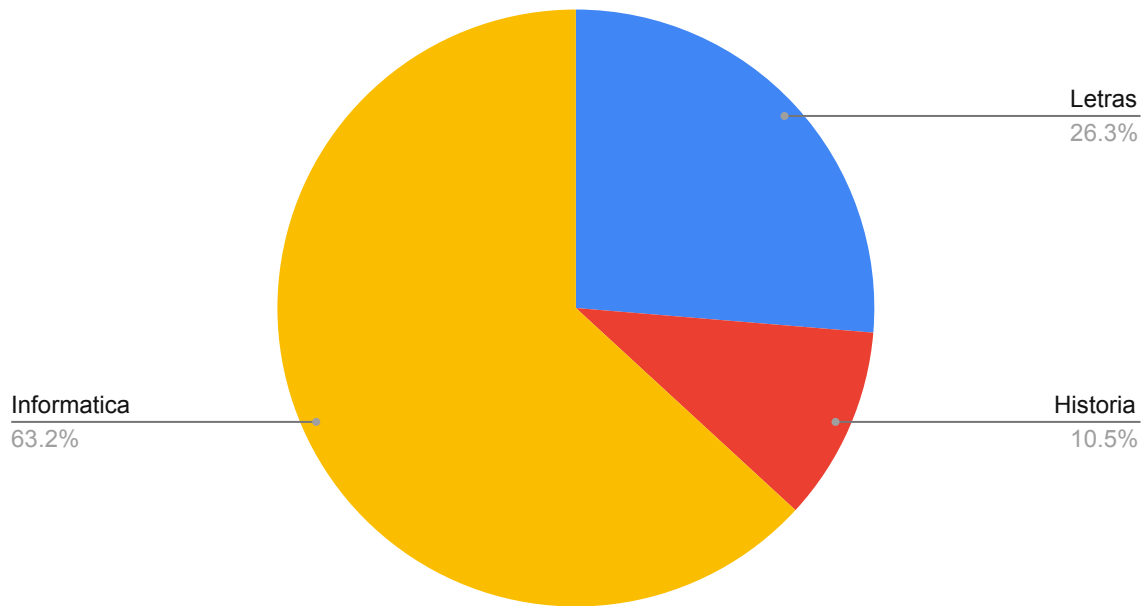


Figura 8.1: Gráfico de composición de especialidades de entrevistados.

Resultados obtenidos

Los 19 cuestionarios enviados por los participantes de la prueba fueron tabulados y analizados separando cada una de las secciones del mismo (test UEQ, preguntas abiertas y test SUS).

Para el test UEQ se totalizaron los valores por funcionalidad y se calculó un promedio de cada una de las escalas correspondiente a los pares de propiedades empleadas. De esta forma se obtuvo un valor entre 1 y 5 el cual podemos definir como nivel de cumplimiento con dicha escala, donde 1 representa el no cumplimiento con la escala y 5 el cumplimiento total con la misma. En base a estos resultados se elaboraron gráficos de barras donde se realiza una comparación de las escalas por funcionalidad.

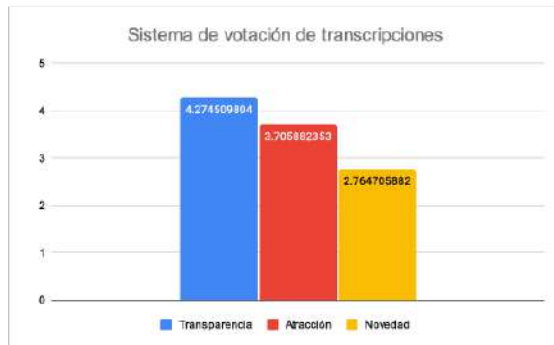
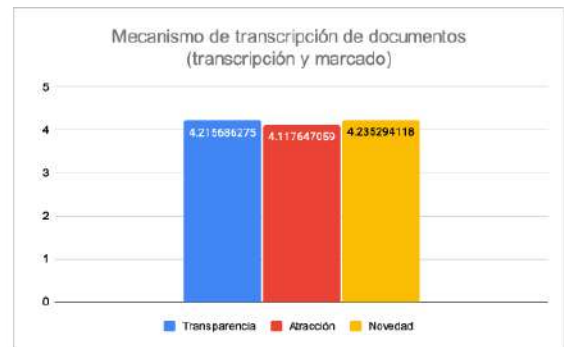
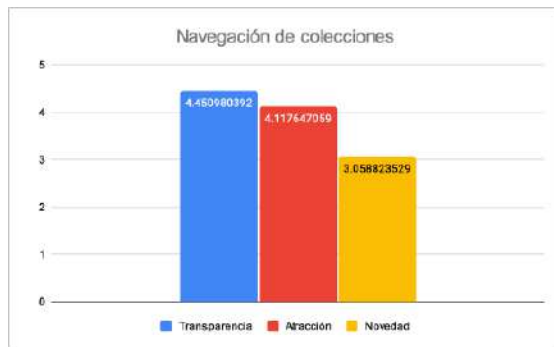


Figura 8.2: Gráficos de escalas UEQ por funcionalidad.

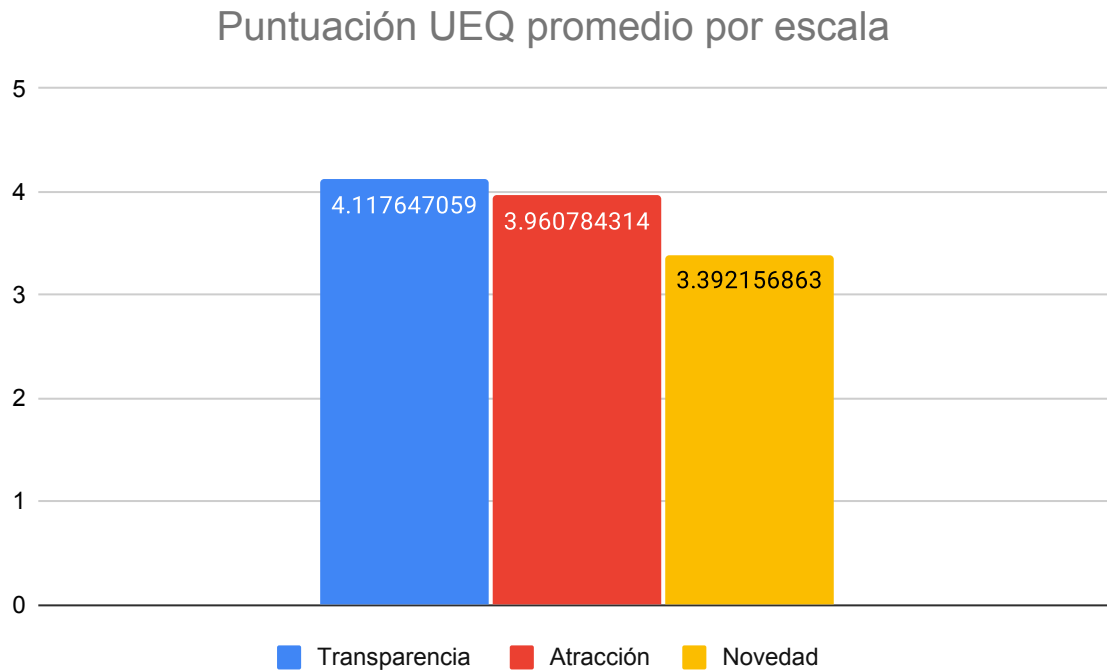


Figura 8.3: Promedio de puntuación UEQ por escala.

Analizando los gráficos de la figura 8.2, se puede notar relaciones entre escalas. Por ejemplo, entre la de novedad y transparencia mientras que en las funciones más comunes de la plataforma (presentes en otros sistemas, como “Foros de discusión”) el índice de novedad es menor al de transparencia. En las funciones mas ligadas a la lógica de negocio como “Mecanismo de transcripción de documento” estas escalas se encuentran más iguales, con menor índice de transparencia, pero con mayor novedad. Esto permite deducir que las características mencionadas inicialmente representan menor innovación y son más fáciles de aprender a usarlas, en cambio, las funcionalidades propias son mas innovadoras pero suponen un desafío mayor para el usuario, al no contar con la experiencia de sistemas anteriores, la curva de aprendizaje sería más pronunciada. En el caso del índice de atracción se sitúa generalmente en un punto intermedio de entre la transparencia y la novedad, indicando una dependencia entre estas escalas y reforzando la idea de “si es fácil de usar y novedoso en cierto aspecto entonces es atractivo”.

Analizando el cumplimiento con los resultados esperados planteados en 8.1.3, se considera a transcriptor una solución innovadora en su campo con un promedio de coeficiente UEQ de novedad de 3.39 (ver figura 8.3) que supera el valor de aceptación para la pregunta 3. Ocurre una situación similar en la pregunta 4, dado que el coeficiente UEQ promedio de atracción fue de 3.96 puntos, se superó el umbral de aceptación confirmando un nivel aceptable de motivación para la colaboración entre usuarios.

En cuanto al test SUS obtuvo el valor promedio total por pregunta y se presentó en un gráfico para facilitar su análisis 8.4. Además, para cada una de las evaluaciones se calculó el puntaje SUS utilizando la formula estándar que define la prueba, estos puntajes se promediaron para poder obtener de esta forma el puntaje SUS final para la plataforma.

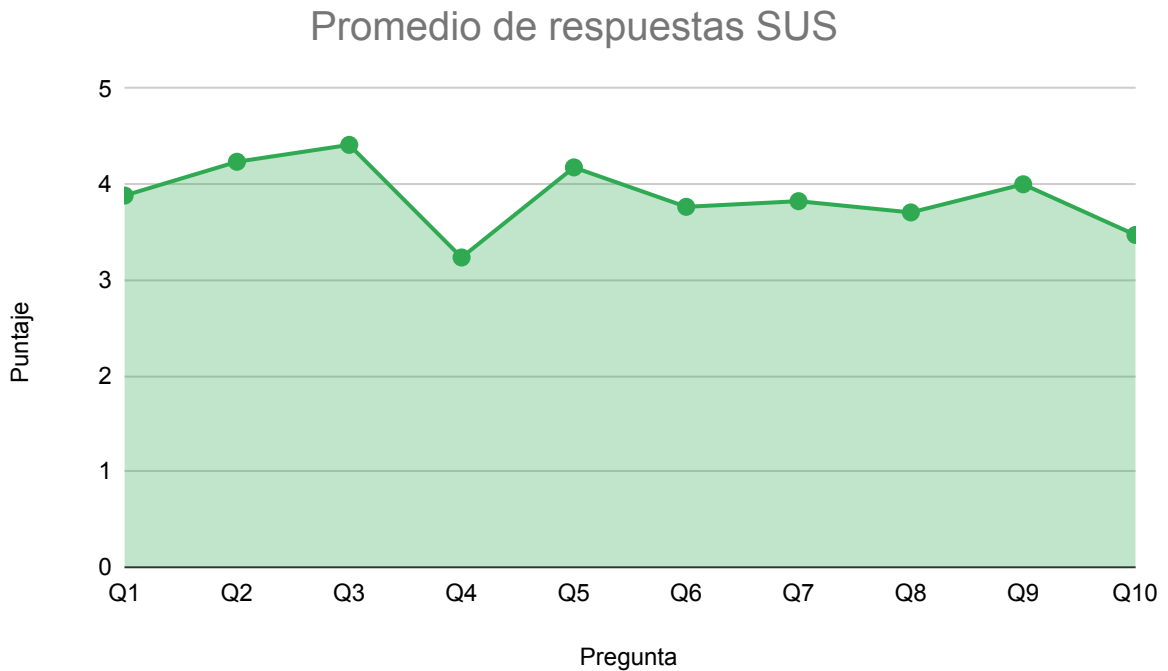


Figura 8.4: Gráfico de resultados SUS por pregunta.

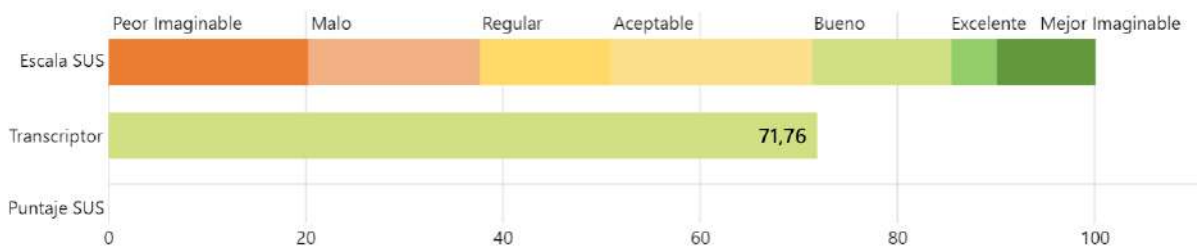


Figura 8.5: Gráfico de puntaje resultado del test SUS.

Como se puede observar en la figura 8.5 el promedio final de puntajes SUS es de 71.6. Bangor, Kortum, y Miller (2008) [23] definen en su estudio una escala basada en siete adjetivos la cual se corresponden directamente con intervalos en la métrica de puntajes utilizada en SUS. Utilizando esta escala de adjetivos se comparó el puntaje obtenido ubicando a la experiencia de usuario de la plataforma en un nivel “Bueno”, lo que asume que la experiencia de usuario se encuentra dentro del 70 % de los sitios analizados en el estudio citado. Esto quiere decir que la experiencia de usuario es notable y que el usuario no tendrá problemas mayores al interpretar el contenido del sitio.

Teniendo en cuenta los resultados esperados planteados en la sección 8.1.3 se concluyó que la plataforma es simple de utilizar y de fácil aprendizaje. Esto se justifica ya que el puntaje SUS promedio superó el umbral de aceptación fijado en 70 puntos. De misma forma sucede con el coeficiente UEQ de transparencia el cual fue de 4.11 (ver figura 8.3) que rebasa el valor de aceptación fijado en 3.

Para las preguntas abiertas se analizó cada de las respuestas y se realizó una puesta en común a cada punto:

■ **¿Encuentra útil la finalidad de transcriptor?**

Los participantes coincidieron que consideran útil la finalidad de transcriptor, haciendo hincapié en la facilidad del uso, la colaboración entre usuarios, lo novedoso del sistema de curado de información y la posibilidad de describir elementos de manera semántica.

■ **¿Piensa que Transcriptor potencia la colaboración entre usuarios?**

Como respuesta unánime los encuestados indicaron la plataforma potencia la colaboración, mencionando además que esta cualidad se debe al marco de trabajo que la herramienta aporta a un proceso repetitivo y complejo como lo es la transcripción de manuscritos.

■ **¿Encontró dificultades al momento de generar transcripciones?**

El 90 % de los entrevistados respondió que no tuvo inconvenientes al utilizar la plataforma. El 10 % restante reportó bugs menores que no les impidió el uso del sistema como formularios que no se limpiaban e inconvenientes para visualizar ciertas líneas de los manuscritos al seleccionarlas con la herramienta “selección de línea”.

■ **¿Encuentra atractivo/intuitivo el mecanismo de transcripción? ¿Cambiarías algo?**

En este punto los usuarios coincidieron en que el mecanismo es atractivo y fácil de utilizar pero en líneas generales necesitaron de una guía para usarlo por primera vez, sobre todo la funcionalidad de transcripción semántica. En este contexto, gran parte propuso establecer un mecanismo de recorrido inicial para presentarles las herramientas y el espacio de trabajo que tienen disponible al momento de realizar una transcripción.

8.1.4. Evaluación de usabilidad mediante recorrido cognitivo

Objetivo

El objetivo de esta evaluación es obtener información acerca del nivel de facilidad del uso y aprendizaje de la plataforma mediante el método de inspección de usabilidad denominado recorrido cognitivo (o Cognitive Walkthrough) el cual consiste en que un grupo de expertos exploren la interfaz de usuarios, realizando acciones previamente definidas para evaluar la facilidad de aprendizaje de un diseño de la plataforma, en este caso, la facilidad de uso de Transcriptor. Para el desarrollo de la prueba se tuvo en cuenta el modelo que define Toni Granollers [24] el cual define tres pasos para la creación de un recorrido cognitivo:

1. Definición de los datos:

- a) Se identifican y documentan las características de los usuarios ¿Quiénes serán los usuarios del sistema? La descripción de los usuarios incluirá la experiencia específica acumulada y el conocimiento adquirido como factores determinantes para la comprobación del factor “cognitivo” durante el recorrido.
- b) Se describe el el prototipo que se utilizara en la evaluación. Este prototipo no es necesario que tenga un alto nivel de detalle.
- c) Se enumeran tareas específicas para realizarse.

- d) Para cada tarea, implementar un conjunto de actividades para llevarlas a cabo.
2. Recorrido de acciones: Los evaluadores realizan cada una de las tareas determinadas anteriormente siguiendo los pasos especificados y utilizando el prototipo detallado. En este proceso, el evaluador utilizará la información del factor cognitivo (experiencia y conocimiento adquirido) de los usuarios para comprobar si la interfaz es adecuada para el mismo. Esta revisión ha de ser minuciosa para todas las acciones especificadas para la consecución de la tarea. Para ello, el evaluador en cada acción criticará el sistema respondiendo a las siguientes preguntas:
- ¿El usuario hará lo correcto en este paso?
 - Si lo hace, ¿sabrá si hizo lo correcto y está progresando hacia la meta?
3. Documentación de resultados:
- El evaluador anotará para cada acción las respuestas del sistema y sus anotaciones.
 - Reporte de problemas de usabilidad (Usability Problem Report Sheet)

En los siguientes apartados se desarrollará la definición de cada uno de estas tareas aplicadas en Transcriptor

Diseño de la prueba (Definición de los datos)

Para aplicar definición de los datos a utilizar mencionados en el punto 1 del modelo de [24], se realizó un análisis del perfil de usuario de la plataforma. Teniendo en cuenta el objetivo y funcionalidades que posee, se puede concluir en que, si bien Transcriptor está disponible para todo el usuario que quiera contribuir en la transcripción de manuscritos, es posible la mayor parte de los perfiles se encuentren ligados a actividades de humanidades digitales, filología o historia. Los cuales son campos que están familiarizados con la carga de formularios y el manejo y carga de documentación digital, así como el manejo de datos semánticos sobre conceptos.

Este arquetipo de usuario se utilizó su vez para establecer la serie de tareas a analizar durante el recorrido. Cada tarea se subdividió a su vez en acciones, las cuales son los pasos a cumplir para completar el objetivo de la tarea.

- Navegar colecciones, proyectos y páginas
 1. Hacer click en el menú “Colecciones”.
 2. Buscar y hacer click en la colección llamada ‘Schmidel’.
 3. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
 4. Buscar y hacer click en la página 1.
- Agregar una nueva capa de transcripción
 1. Hacer click en el selector de capas en la esquina inferior izquierda.
 2. Hacer click en “Agregar capa”.
 3. Elegir un nombre de su interés y hacer click en guardar.
- Realizar una transcripción básica

1. En la campo situado a derecha de la pantalla, escribir algo de lo que lea en la imagen del manuscrito.
 2. Al texto ingresado, darle formato al primer párrafo con el panel superior, buscar la opción “negrita”.
 3. Hacer click en guardar.
- Realizar una transcripción con marca
 1. Buscar en los controles de la imagen la opción “dibujar marca”.
 2. Hacer click en donde comienza el primer renglón de la imagen y hacer click en el final del mismo.
 3. En la ventana emergente escribir lo que se alcanza a leer en el área seleccionada.
 4. Revisar lo escrito y hacer click en guardar la transcripción.
 - Realizar una transcripción Semántica
 1. Hacer click en el selector de capas en la esquina inferior izquierda.
 2. Hacer click en “Agregar capa”.
 3. Escribir como nombre de capa “Firma” y hacer click en guardar.
 4. Seleccionar la capa “Firma”.
 5. Buscar en los controles de la imagen la opción “dibujar marca”.
 6. Marcar el área donde se encuentra la firma.
 7. Elegir la ontología FOAF en el panel derecho de la pantalla y click en el botón siguiente,
 8. Elegir el datatype Person y hacer click en el botón siguiente.
 9. Hacer click en botón .Agregar propiedadz seleccionar las data properties Name y Surname y hacer click en el botón Çerrar”.
 10. Llenar los campos Name y Surname con los datos: Juan y Perez respectivamente y hacer click en el botón siguiente.
 11. Click siguiente sin agregar ninguna relación
 12. Hacer click en guardar.
 - Publicar un comentario en el foro de discusión
 1. Hacer click en la marca que aparece en la imagen del documento.
 2. Hacer click en el botón llamado “Foro”.
 3. Escribir un comentario y hacer click en botón guardar.
 - Votar por una transcripción
 1. Buscar y hacer click en una marca.
 2. Hacer click en el botón otras transcripciones.
 3. Buscar una transcripción y hacer click en el botón “Me gusta”.
 - Exportar los datos de una colección
 1. Hacer click en el menú “Colecciones”
 2. Buscar y hacer click en la colección llamada ‘Schmidel’
 3. Hacer click en el botón de configuración
 4. Hacer click en el botón exportar colección

Desarrollo de la prueba (Recorrido de acciones)

Para el desarrollo de la prueba (paso 2 del modelo de Toni Granollers [24]) se montó un servidor con la plataforma instalada en la cual se cargaron varios documentos para su transcripción. En este contexto, se organizó una reunión entre especialistas en informática y se recorrió en conjunto el sistema realizando las tareas definidas en la sección 8.1.4. Para cada acción que compone la tarea se respondió el par de preguntas dadas por el modelo, estas son:

1. ¿El usuario hará lo correcto en este paso?
2. Si lo hace, ¿sabrá si hizo lo correcto y está progresando hacia la meta?

A medida que los datos de cada respuesta eran obtenidos se registraban para su posterior análisis.

Resultados obtenidos (Documentación de resultados)

Partiendo de los datos obtenidos en el proceso desarrollado en la sección 8.1.4 se confeccionó una tabla con las respuestas dadas para simplificar el análisis de las mismas. Dicho análisis tiene por objetivo detectar cualquier potencial problema de usabilidad en las funciones recorridas. En función a este estudio se elaboró un reporte con los problemas de usabilidad encontrados clasificándolos por severidad y proponiendo una posible solución.

Tarea	Acción	Problema	Severidad	Posibles Soluciones
Transcripción con marca	Hacer click en donde comienza el primer renglón de la imagen y hacer click en el final del mismo.	Dependiendo del texto es complejo para el usuario seleccionar correctamente el inicio y/o final del renglón y la marca quede dibujada en una sección incorrecta del manuscrito	1 (baja)	1) Añadir una grilla de referencia sobre el manuscrito para que actúe como una guía de marcado de los renglones. 2) Reconocer automáticamente los renglones a través de procesamiento de imágenes para que de esta forma el usuario haga click sobre los mismos para cargar la marca.
Transcripción básica	Guardar cambios	Si el usuario olvida hacer click en guardar los cambios de la transcripción, al salir de la pantalla de transcripción se perderá el progreso que llevaba hasta el momento.	3 (media)	1) Agregar una confirmación al salir de la pantalla de transcripción si tiene cambios sin guardar. 2) Notificar al usuario que tiene cambios sin guardar. 3) Como alternativa, implementar un guardado automático de las modificaciones cada determinada cantidad de tiempo.

8.2. Presentaciones

8.2.1. II Congreso de Ciencia Abierta y Ciudadana en Argentina (CIACIAR)

En noviembre de 2018 llevó a cabo el II Congreso de Ciencia Abierta y Ciudadana en Argentina (CIACIAR) en la Universidad Nacional de San Martín (UNSAM). El mismo tuvo por objeto conectar y dar cuenta de las diferentes experiencias, prácticas y actores que hacen ciencia abierta y ciudadana con el fin de desarrollar iniciativas que permitan consolidar el proceso de apertura del conocimiento científico. Transcriptor como sistema basado en el crowdsourcing e interacción de una comunidad para lograr los objetivos que propone puede ser considerado como proyecto que hace uso de la ciencia ciudadana, encajando en el perfil del congreso. En este contexto se realizó una presentación de Transcriptor en modalidad de feria de ciencias. Se puso a disposición de los participantes computadoras con la plataforma para que la navegaran y realizaran transcripciones de digitalizaciones precargadas. Además, se realizó la presentación de un póster informativo sobre los flujos de uso de la plataforma, roles y tratamiento de la información.

8.2.2. III Congreso Internacional de la Asociación Argentina de Humanidades Digitales (AAHD)

La Asociación Argentina de Humanidades Digitales (AAHD) realizó en la Facultad de Humanidades y Artes de la Universidad Nacional de Rosario realizó en noviembre de 2018 un congreso donde se trataron varios tópicos relacionados al proceso de digitalización de las humanidades. Haciendo hincapié en los recursos educativos, métodos de colaboración, proyectos y tecnologías, herramientas digitales, preservación de la información, entre otros. En este marco se realizó la presentación de Transcriptor en el área “Culturas participativas y colaborativas”, donde se habló sobre el objetivo de Transcriptor como sistema, mostrando el esquema de trabajo y debatiendo con especialistas de la materia la importancia que una herramienta digital colaborativa aporta al área de humanidades.

Transcriptor

Integrantes
Bracco, Christian
Cuevas, Lucas
Del Río Riande, Gimena
Torres, Diego

Sistema colaborativo para la transcripción de manuscritos históricos

Imaginate que tenés un montón de cartas o documentos manuscritos viejos. Que además, esas cartas las escribieron personas importantes en la historia, como por ejemplo la carta de Manuel Belgrano que está en la foto.

Ahora imaginate que además queremos que el texto de esas cartas pueda encontrarse usando un buscador por Internet para que sea utilizado en las escuelas o en otro lugar.

Transcriptor es un proyecto que busca crear una plataforma de ciencia ciudadana para poder realizar transcripciones de documentos que fueron digitalizados con una foto y que no han podido ser transcritos en forma automática. De manera que puedas copiar, pegar, buscar, consultar y editar contenidos escritos hace décadas!

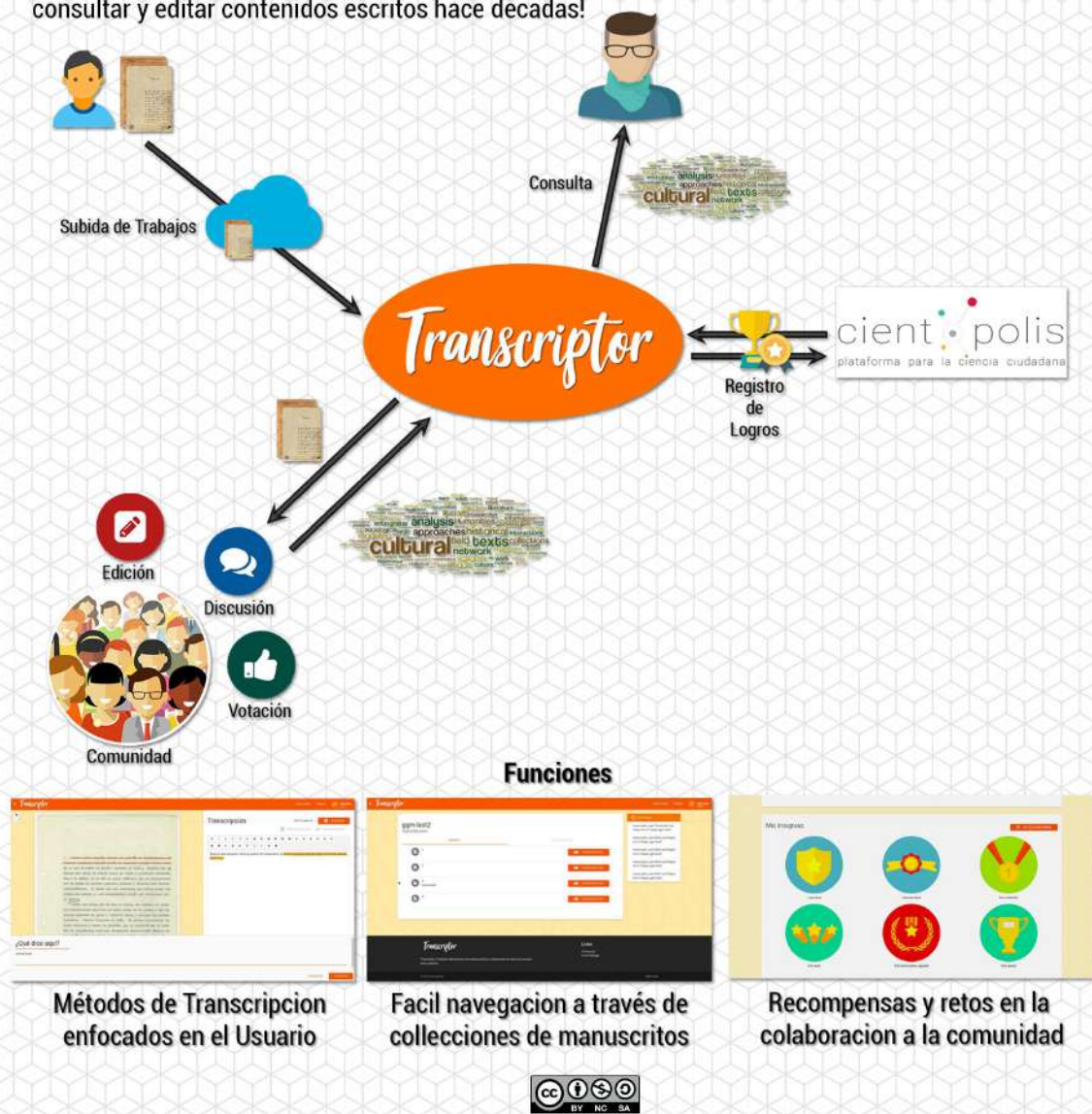


Figura 8.6: Póster presentado en CIACIAR 2018.

8.2.3. #NoviembreHD

En noviembre de 2020 debido a la pandemia de COVID-19 y con el crecimiento de las herramientas de videoconferencias la Asociación Argentina de Humanidades Digitales (AAHD) decidió organizar un evento online el cual enmarcaba diferentes actividades sincrónicas y asincrónicas, desde paneles, vídeos, demos y pósteres hasta encuentros, paneles y talleres. Las actividades pudieron realizarse a través de Twitter, Wikipedia y otras plataformas de reuniones virtuales como Jitsi, Meet y Zoom. Se realizó una presentación de Transcriptor a través de Twitter iniciando un hilo con un póster digital al cual se le adicionó en la publicación una breve explicación del objetivo y funcionamiento de la plataforma. Conforme a los participantes visualizaban el ePoster, tenían opción de responder al hilo publicando sus dudas o sugerencias, las cuales luego eran respondidas a través del mismo medio. Se respondieron dudas sobre el modelo de trabajo, interacción entre usuarios y curado de la información.

8.2.4. III Congreso de Ciencia Abierta y Ciudadana en Argentina (CIACIAR)

Transcriptor fue presentado en una nueva edición del CIACIAR, en este caso y por medidas de distanciamiento en función a la pandemia de COVID-19, se realizó de manera virtual. Los distintos paneles se realizaron a través de zoom, youtube y twitter donde se presentaron multiples proyectos relacionados con la ciencia ciudadana. Como introducción a la plataforma se utilizó un ePoster (gif de 5 diapositivas de una presentación) y se realizó una presentación a través de zoom, la cual se transmitió en vivo por youtube. Durante la charla, se habló de las distintas modalidades de transcripción junto al flujo de la información dentro del sistema (carga de documentos, transcripción, exportación de los datos).

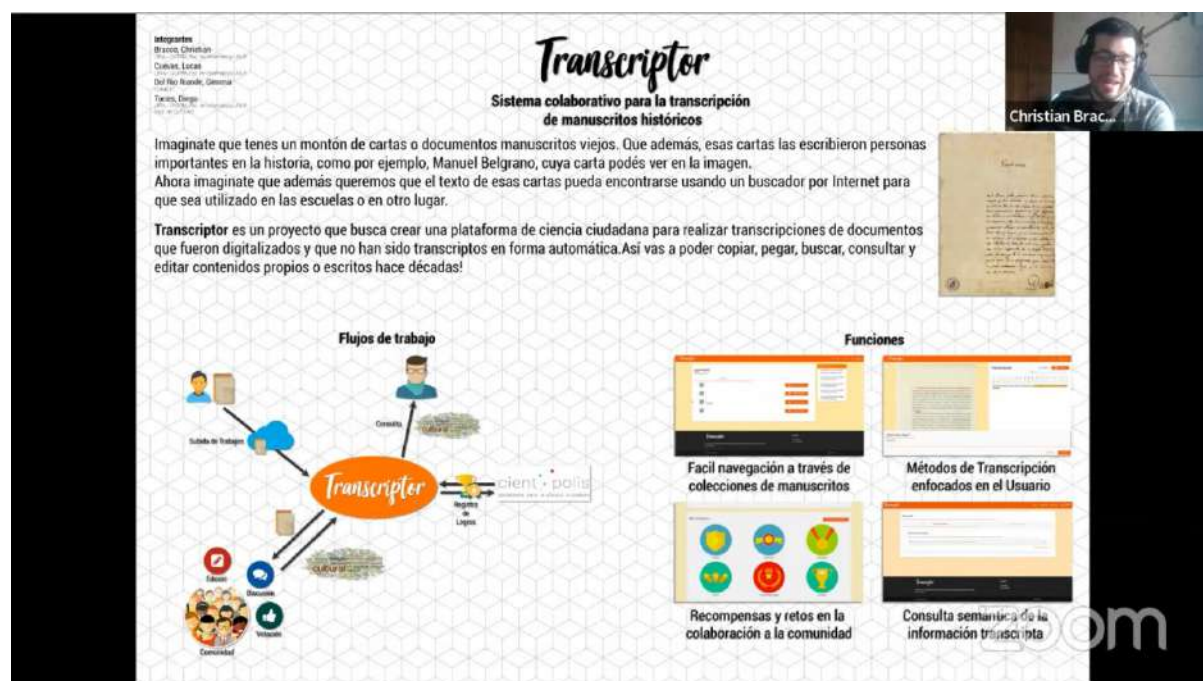


Figura 8.7: Captura de la transmisión de la presentación durante el CIACIAR 2021.

8.2.5. HCI International 2023

Transcriptor fue presentado en HCI (Human-Computer Interaction, HCI) International 2023 la cual es una conferencia Internacional sobre Interacción Humano-Computadora. Este evento es conocido por ser un foro multidisciplinario que reúne a profesionales, investigadores y académicos interesados en los aspectos teóricos y prácticos de la interacción humano-computadora. La presentación de Transcriptor fue realizada mediante un paper titulado: “Semantic annotation in collaborative document transcription: a gamified citizen science approach” (ver apéndice [A](#)) que expone las características principales de Transcriptor tales como la transcripción manual y la transcripción semántica y las evaluaciones de usabilidad que se realizaron junto con los resultados obtenidos.

Capítulo 9

Conclusión y trabajo futuro

La transcripción de manuscritos es una tarea ardua y meticulosa que se extiende mucho más allá de una simple transliteración del texto de una fuente a otra. Requiere de un proceso de definición, comprensión y organización de los contenidos que, debido a la falta de métodos automatizados confiables, le agrega complejidad y extiende sus tiempos de realización. Facilitar este trabajo unificando las principales tareas en una plataforma fue el puntapié inicial para el desarrollo de Transcriptor.

Diseñar y construir esta plataforma representó un desafío de descubrimiento, análisis y optimización de la labor de los transcritores (filólogos, historiadores, especialistas en letras) haciendo hincapié en las necesidades que actualmente experimentan, generando un espacio de trabajo con un enfoque colaborativo entre miembros de una comunidad.

Actualmente transcriptor brinda un flujo de trabajo sencillo, que permite a interesados de todos los niveles educativos colaborar en un proyecto, visualizar, votar y comentar las transcripciones de otros usuarios. Esto plantea una solución eficiente frente a los problemas en el reconocimiento automatizado del texto manuscrito impulsando iniciativas como el crowdsourcing y la ciencia ciudadana.

Otro inconveniente planteado inicialmente es la pérdida de información que sufre un manuscrito al momento de ser transcrito. Contexto posicional y elementos gráficos que no pueden ser representados en un simple texto plano contribuyen a que el lector nunca se percate de detalles que solo podrían notarse visualizando la página original.

Para atacar este problema, Transcriptor realiza aportes novedosos al proceso de transcripción, enriqueciendo la información obtenida con la incorporación de referencias posicionales entre el texto transcrito y la página digitalizada. La adición de un sistema de anotaciones semánticas también es una característica crucial para llevar la transcripción a otro nivel, permitiendo al colaborador ya no solo transcribir el contenido textual de un manuscrito, sino describir elementos presentes en el documento original que no pueden ser transliterados textualmente (imágenes, sellos, marcas y cualquier elemento con valor pictórico). Este sistema de transcripción también permite incorporar información a la información ya cargada (metaformación), brindando la posibilidad de anotar conceptos y términos particulares, relacionándolos con datos ya existentes en la base de conocimiento de Transcriptor.

La gamificación integrada a los flujos de trabajo actúa como recurso efectivo para motivar al usuario a realizar aportes. Cada acción dentro de la plataforma se registra y contabiliza para incrementar el rango de experiencia del usuario. De esta manera el transcriptor se esfuerza para realizar mayor cantidad de colaboraciones en los proyectos para avanzar en su plan de carrera.

Teniendo en cuenta los puntos y los objetivos planteados inicialmente así como las eva-

luaciones realizadas sobre la plataforma, se considera que Transcriptor es una solución efectiva a los problemas actuales de la transcripción de manuscritos. Transcriptor logra generar un espacio de trabajo en el que personas de cualquier índole, ya sea investigadores, especialistas o aficionados puedan colaborar en proyectos de transcripción reales impulsados por organizaciones del ámbito (bibliotecas, escuelas, universidades, ministerios). Esto permite el intercambio de conocimientos entre los participantes, beneficiando tanto a los responsables del proyecto como a cada colaborador, enriqueciéndose en la experiencia y estimulando el aprendizaje continuo en la colaboración y contenidos que se transcriben, brindando a la comunidad material más sencillo de leer, buscar y utilizar sin perder la esencia y detalles presentes en un manuscrito original.

9.1. Trabajo futuro

Actualmente Transcriptor logra cumplir el propósito para el que fue diseñado presentando una solución óptima a los problemas planteados. Aún así, en este contexto es posible continuar mejorando el producto extendiendo o cambiando el enfoque de ciertas funcionalidades. A continuación, agrupadas por tópico, se presentan posibles ideas para extender esta tesina.

9.1.1. Mejoras en la herramienta de transcripción

- Moderación de aportes y Control de acceso a proyectos. Transcriptor es una plataforma de acceso público, cualquier persona puede registrarse y colaborar con la transcripción de un trabajo. Este beneficio a su vez lo vuelve vulnerable a usuarios malintencionados, que pueden "interferir" en el significado real del documento a transcribir, teniendo la posibilidad de ingresar transcripciones no válidas. Agregar un mecanismo de control de acceso a los propietarios de cada proyecto podría representar una solución a estas situaciones. También podría analizarse la moderación de aportes, realizada por usuarios con cierto rango o perfil para que requieran una revisión antes que se publiquen.
- Características Tiempo Real. Una funcionalidad útil en la confección de documentos en colaborativo es visualizar las modificaciones que otros colaboradores realizan en tiempo real. Esto facilita el seguimiento de las tareas para evitar inconvenientes de superposición de párrafos entre usuarios o simplemente para prestar ayuda en ciertos pasajes del texto. Adicionalmente también se podría incorporar un chat abierto por documento para que los colaboradores puedan interactuar y ponerse de acuerdo con su punto de vista para las transcripciones.
- Transcripción asistida por Métodos de Reconocimiento. Si bien el objetivo de esta tesina es presentar una alternativa colaborativa ante las inconsistencias de los mecanismos automáticos de reconocimiento de texto manuscrito, es imposible negar que los avances en este campo presentan resultados con una tasa de errores cada vez más baja. En este contexto podrían aprovecharse las bondades del reconocimiento de texto y la colaboración para elaborar un esquema mixto en el cual los métodos automáticos asistan la transcripción que el colaborador está realizando. Por ejemplo, en pasajes donde el texto sea más regular aplicar métodos de reconocimiento como OCR para generar sugerencias y que el usuario decida cuál es la correcta. Otro ejemplo podría aplicarse a la cantidad de renglones que tiene una página, permitiendo establecer un total sobre el cual plantear un porcentaje de avance sobre el documento.

9.1.2. Aplicación Móvil

La arquitectura de Transcriptor se basa en una aplicación backend la cual expone sus servicios a través de una API Rest a un Frontend Web (capítulo 7.1). Esta arquitectura a su vez facilita la incorporación de clientes que hagan uso de estos servicios para integrar la solución a nuevos tipos de dispositivos o plataformas. El desarrollo de una aplicación móvil, con funciones básicas como transcribir y realizar marcas permitirá extender el alcance de la aplicación dando lugar a que mas colaboradores tengan acceso. Además el hardware presentado por los teléfonos actuales simplificarían ciertas tareas dentro de la plataforma, por ejemplo, la pantalla táctil de los smartphones facilitaría el marcado de las líneas de texto a transcribir.

9.1.3. Mejoras en la herramienta de transcripción semántica

- Conectar y compartir datos con otras ontologías: En este punto, Transcriptor conforma un vocabulario aislado de la red abierta de datos enlazados (Linked Open Data). Un buen aporte sería conectar el modelo de datos de la plataforma para que sea accesible y utilizable por otros vocabularios. Esta mejora permitiría referenciar elementos de Transcriptor desde entidades que existan en otras bases de conocimiento, como DBpedia.
- Consumir información desde otros vocabularios: En complemento al punto anterior, podría extenderse el sistema para enriquecer los datos de las entidades de Transcriptor enlazando elementos de otros vocabularios. Por ejemplo, Asociar una marca a un autor que este guardado en Wikidata.
- Agregar nuevos formatos de exportación para los proyectos, trabajos y paginas: Actualmente los proyectos solo pueden ser exportados en formato RDF, el cual es estándar para representar datos semánticos. Un desafío sería evaluar e integrar otros formatos exportables, estos podrían ser estructurados, como XML, JSON u OWL, que permitirían importarlos y/o procesarlos en otros sistemas, o formatos de lectura como PDF, Markdown o DOC que sean fácilmente interpretados por personas.

9.1.4. Extender gamificación

La gamificación implementada en Transcriptor representa un acercamiento básico a la motivación y engagement del usuario, esta se ve reflejada en las tareas básicas como iniciar sesión, colaborar y subir trabajos a transcriptor que suman puntos para que el colaborador incremente su rango. La aplicación de un estudio a mayor profundidad sobre como afectan en el usuario estas las recompensas de acción dentro de Transcriptor podría generar nuevas alternativas de motivación y ayudar a extender el enfoque actualmente implementado. Ideas como la definición de nuevos premios, actividades alternativas para ganar recompensas, mayor cantidad de niveles alcanzables, extender la ruta de progreso del jugador e incluso desbloquear funcionalidades de Transcriptor según su rango podrían ser puntos de partida para aumentar la calidad del producto en este ámbito.

Bibliografía

- [1] C. Huitfeldt and C. Sperberg-McQueen, "What is transcription?," *Literary and Linguistic Computing*, vol. 23, pp. 295–310, 09 2008.
- [2] "Ocr, icr y omr. ¿qué son y para qué sirven?," Nov 2016.
- [3] M. J. Brown, "Experiments with early modern manuscripts and computer-aided transcription," Feb 2019.
- [4] M. Moyle, J. Tonra, and V. Wallace, "Manuscript transcription by crowdsourcing: Transcribe bentham," *Liber Quarterly*, vol. 20, p. 3, 01 2010.
- [5] B. Tanodi, "Documentos históricos. normas de transcripción y publicación," *Cuadernos de Historia. Serie economía y sociedad*, no. 3, pp. 259–270, 2000.
- [6] L. Eikvil, "Optical character recognition," *citeseer.ist.psu.edu/142042.html*, 1993.
- [7] D. Deodhare, N. R. Suri, and R. Amit, "Preprocessing and image enhancement algorithms for a form-based intelligent character recognition system.," *IJCSA*, vol. 2, no. 2, pp. 131–144, 2005.
- [8] A. Juan, V. Romero, J. A. Sánchez, N. Serrano, A. H. Toselli, and E. Vidal, "Handwritten text recognition for ancient documents," in *Proceedings of the First Workshop on Applications of Pattern Analysis*, pp. 58–65, 2010.
- [9] M. M. Terras, "Crowdsourcing in the digital humanities," in *Companion to Digital Humanities II*, Wiley-Blackwell, 2016.
- [10] H. C. Sáez *et al.*, "Ciencia ciudadana, ética y política para viejos y nuevos problemas," *Biblio 3w: revista bibliográfica de geografía y ciencias sociales*, 2014.
- [11] P. Castells, "La web semántica," *Sistemas interactivos y colaborativos en la web*, pp. 195–212, 2003.
- [12] R. Pedraza, L. Codina, and C. Rovira, "Web semántica y ontologías en el procesamiento de la información documental," *El profesional de la información. 2007; 16 (6): 569-78*, 2007.
- [13] J. C. Tello, "La web semántica y el lenguaje rdf," 2006.
- [14] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee, "Linked data on the web (ldow2008)," in *Proceedings of the 17th international conference on World Wide Web*, pp. 1265–1266, 2008.
- [15] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and N. Lindström, "Json-ld 1.0," *W3C recommendation*, vol. 16, p. 41, 2014.

- [16] G. Zichermann and C. Cunningham, *Gamification by design: Implementing game mechanics in web and mobile apps*. O'Reilly, 2011.
- [17] K. Werbach and D. Hunter, *Gamificación: Revoluciona Tu Negocio con las técnicas de los juegos*. Pearson Educación, 2014.
- [18] O. Borrás Gené, "Fundamentos de gamificación," *Gabinete de Tele-Educación de la Universidad Politécnica de Madrid*, 2015.
- [19] A. Aguilera Castillo, C. A. Fúquene Lozano, and W. F. Ríos Pineda, "Aprende jugando: el uso de técnicas de gamificación en entornos de aprendizaje," *IM-Pertinente*, vol. 2, no. 1, pp. 125–143, 2014.
- [20] D. D. R. Robert Simpson, Kevin R. Page, "Zooniverse: Observing the world's largest citizen science platform," *Literary and Linguistic Computing*, pp. 295–310, 09 2008.
- [21] S. D. Pravisani, *Metagame: Un juego de Ciencia Ciudadana*. PhD thesis, Universidad Nacional de La Plata, 2017.
- [22] P. Kortum, C. Z. Acemyan, and F. L. Oswald, "Is it time to go positive? assessing the positively worded system usability scale (sus)," *Human factors*, vol. 63, no. 6, pp. 987–998, 2021.
- [23] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.
- [24] T. Granollers and J. Lorés, "Incorporation of users in the evaluation of usability by cognitive walkthrough," in *HCI related papers of Interacción 2004*, pp. 243–255, Springer, 2006.

Parte III

Anexo

Apéndice A

Documento presentado en HCI International 2023

Semantic annotation in collaborative document transcription: a gamified citizen science approach

Christian Ezequiel Bracco¹, Lucas Ezequiel Cuevas¹, and Diego Torres^{1,2}[0000-0001-7533-0133]

¹ LIFIA, CICPBA-Facultad de Informática, UNLP,
50 y 120, 1900 La Plata, Argentina
diego.torres@lifia.info.unlp.edu.ar

² Departamento de Ciencia y Tecnología, UNQ
Roque Sáenz Peña 352, Bernal, Argentina

Abstract. Human or manual transcription is the task in which a person reads a handwritten document and writes in a digital environment the text they are reading. The manual transcription task is a long and time-consuming process for one person. However, the intelligence of the transcriber is provided with better results than automatic alternatives. This article introduces a collaborative transcription platform called Transcriptor, in which the community members can upload digitized manuscripts and collaborate in transcribing them, defining different transcription layers which will be represented using semantic web technologies and gamification techniques. The article provides two experimentation that shows that Transcriptor had a good acceptance and novelty.

Keywords: Transcription · Citizen Science · Semantic Web

1 Introduction

Documents digitization has become a fundamental tool for preserving, providing, and extending access to archival collections. Libraries and cultural heritage institutions provide collections of rare books, manuscripts, and old photographs that have deteriorated over time, and a physical manipulation could compromise the existence of the cultural artifact[6]. However, converting content images into machine-readable data that can be searched, sorted, and manipulated opens up a new range of difficulties and problems to be solved.

Nowadays, the automatic conversion of this data could be improved with many tools to transcribe texts digitized in static images to editable text. One of the most relevant approaches is Optical Character Recognition (OCR) [11], which has advanced significantly over the years, becoming more and more reliable and intelligent. However, the results on handwritten text are still not completely favorable regarding the diversity of languages, and handwriting styles [10].

Handwritten documents, moreover, often present marks made during and after the time of their composition (such as corrections, erasures, crossings out, calls, stamps, or marks) or details typical of the passage of time (wear, stains),

which make their legibility and interpretation more difficult. Indeed, the marks in the documents could generate a wrong transcription because such mark gives the text a context (historical, circumstantial, or emotional) in which it was written, generating situations in which these marks determine the document's meaning. In addition, other elements are related to the structure of the handwritten document: tables, lists, text in columns, text in an unconventional order, and a combination of languages. Therefore, the recognition of these types of texts remains a challenge, and depending on human supervision to ensure the quality and reliability of the obtained texts [12], the idea of using automatic recognition techniques is discarded.

Human or manual transcription is the task in which a person reads a handwritten document and writes in a digital environment the text they are reading. The manual transcription task is a long and time-consuming process for one person. However, the intelligence of the transcriber is provided with better results than automatic alternatives [2]. The semantic web[3, 15] provides technical elements that make data readable by a computer and consequently apply tools to knowledge discovery. There are initial approaches to citizen science and crowd-sourcing task forces. For example, From the Page³ is a crowd-sourcing product to transcribe, collaborate and manage transcriptions. Similarly, the Scribe Project[17] is an open-source framework created by Zooniverse[16] in conjunction with the New York Public Library, allowing easy setup and launch of crowd-sourcing transcription projects, mainly aimed at obtaining structured data from handwritten materials. However, none includes a semantic web description layer, the capacity to improve the differences among transcribers by a discussion or voting element, and a gamification approach to motivate participation and engagement among projects and users.

The approaches require the availability of a space where the work of a group of people interested in the interpretation and collaboration of transcription can be organized and the necessary tools to carry it out. A citizen science model is a suitable approach for this transcription challenge. Citizen science is scientific research carried out by a sum of collaborators (scientists and professionals together) with regular people, which, in this case, is exercised in the participation of a community in handwritten transcription projects. In a collaborative approach to transcriptions, we should consider how to maintain interest and encourage people to collaborate on the projects.

This article introduces a collaborative transcription platform, Transcriptor, in which the community members can upload digitized manuscripts and collaborate in transcribing them (Shown in Figure 1), defining different transcription layers which will be represented using semantic web technologies. Furthermore, this work includes gamification elements in transcriptions so that participants face challenges in games, meet goals and thus obtain achievements and badges, enhancing and making the process more enjoyable. Also, Transcriptor includes voting alternatives defined by each transcribed phrase (Figure 2). Finally, the

³ <https://fromthepage.com> accessed on 8th february 2023.

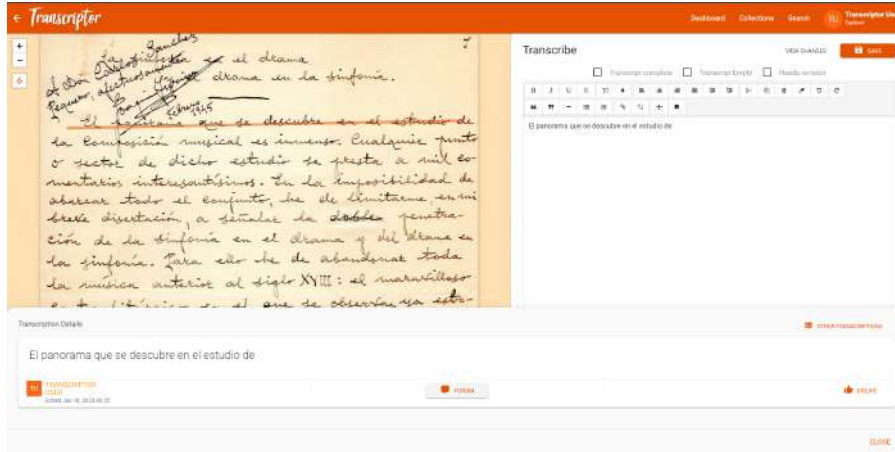


Fig. 1. Screen of Transcriptor in action.

article also includes a usability and experience user evaluation by analyzing 19 recorded user interaction activities with Transcriptor.

The organization of this article is the following. Section 2 describes other related approaches on document transcriptions. Then, Section 3 introduces the main functionality of Transcriptor detailing the process of making a transcription with different alternatives and a brief development details. Then, an evaluation is carried out in Section 4. Finally, conclusions and further work are introduced in Section 5.

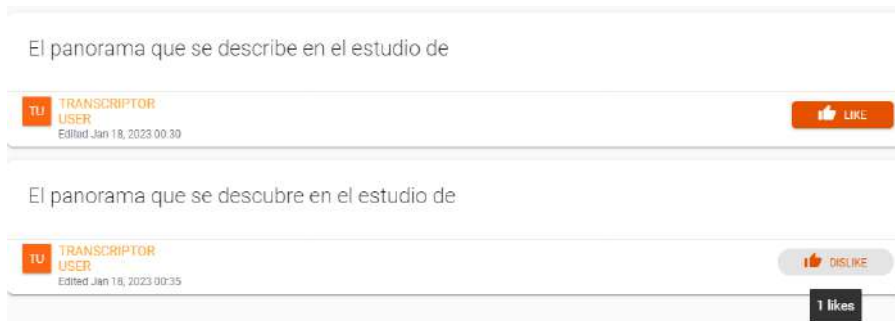


Fig. 2. Votes in Transcriptor.

2 Related Work

FromThePage[13, 14] is an open source collaborative transcription and translation website for digitized documents where community members can upload manuscripts as images or PDFs and collaborate in the transcription or translation of these documents providing the user with a simple environment to do this work. FromThePage provides the basic functionality to Transcriptor. However, Transcriptor extends with more functionality related to gamification and the use of semantic annotations.

The Scribe Project is an open source framework⁴, created by both Zooniverse⁵ and the New York Public Library⁶, which allows working on crowdsourcing transcription projects, particularly aimed at obtaining structured data from handwritten materials or those materials that are not accessible by OCR.

3 Transcriptor

Transcriptor is a web application that allows users to manage and transcribe digitized documents using collaborative editing tools. Transcriptor’s basic functionalities include managing document collections, organizing collaborative transcription projects and transcribing documents by page.

Transcriptions are performed using a collaborative textual typing tool in combination with a photographic document visualization tool. Transcription is performed at different levels allowing to transcribe information from manuscripts, images, and any photographed object. In Transcriptor, transcription is not only performed in terms of text but can also include semantic descriptions for better use of automatic processing.

Transcription Environment The Transcription environment, or transcription screen of a manuscript, is the main section within Transcriptor; this is because this is where the transcription information is materialized and generated collaboratively and where the transcription flow is accomplished.

When entering the transcription environment, several components are distinguished (see Figure 3):

- **1 Document viewer:** allows viewing the digitized document. It is possible to zoom in on the different fragments of the document, navigate it and make marks.
 - **1.1 Zoom controls:** allows controlling the zoom level on the document.
 - **1.2 Drawing Controls:** allows entering in markup mode. Allows to draw polygons over the areas to transcribe. The polygon is then transcribed as a unit.

⁴ <http://scribeproject.github.io> , accessed on 8th february 2023

⁵ <http://zooniverse.org/>

⁶ <https://www.nypl.org/>

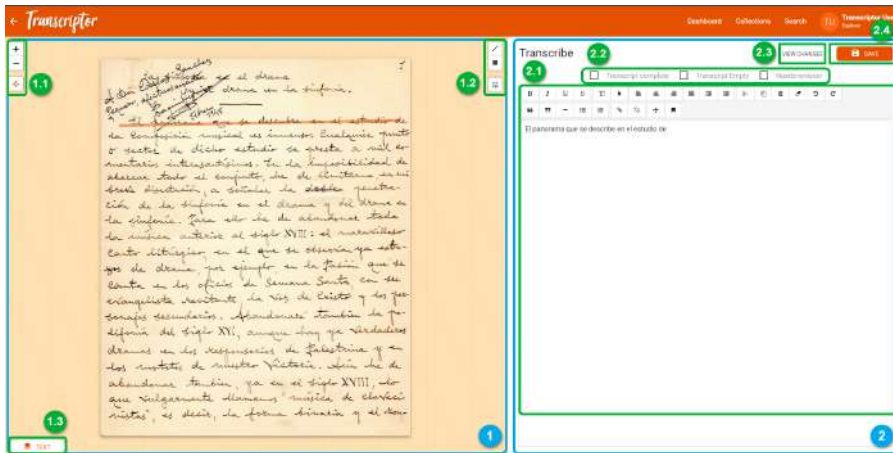


Fig. 3. Transcription environment

- **1.3 Image Adjustments:** enables controls to change the brightness, contrast, and saturation of the digitized image to facilitate the visualization of the written text.
 - **1.4 Layer selector:** opens the transcription layer selection menu; from this menu it is also possible to access the layer management screen, which allows you to modify or delete layers.
- **2. Transcription Panel:** It contains the user interaction logic with that of the transcription state.
- **2.1 Text Editor:** Component destined to the entry and preview of the final text allows selecting regions of the reader to reference them to document sections. Being a WYSIWYG text editor, it has functions to apply styles to the text, bold and underlined, among others.
 - **2.4 Save Button:** It executes the saving of the current state of the transcribed text. It also generates a new version. The “View Changes” button accesses the version menu of the manuscript transcription (2.3).
 - **2.4 Status selector:** allows you to mark the page as “empty” or “pending revision.”

3.1 Transcription and text generation elements

Transcriptor models transcriptions using relationships between its entities, seeking not only to represent the textual content of a manuscript but also to enrich it by associating it directly with the original document context, thus maintaining a reference between fragments of the document and the transcribed text.

Marks are entities that model a line or polygon which selects a portion of the digitized document. They have all the necessary elements for the client to

render it on the manuscript, among them, a set of coordinates relative to its digital image, which are used to be positioned and drawn in the viewer.

As for the text, two representations are handled, one directly linked to the mark, which we call Transcription, which describes the textual content transcribed by the user for a mark on the document, and another representation that is oriented to maintain the structure and overall progress of the transcription process called Transcription Template.

Transcription Templates are “Maps” of relationships used to reference the order in which the text of the marks is combined, allowing to dynamically generate the complete text of the document to be transcribed. Figure 4 shows templates transcription.

3.2 Transcription Strategies

To generate the complete transcription of the manuscript, Transcriptor implements multiple strategies to achieve this, providing the user with several ways to organize his work logic while maintaining the same objective. The transcription strategies supported by Transcriptor are:

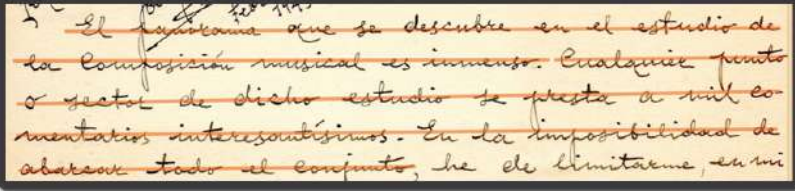
- **Mark to load:** The user selects an area of the digitized document generating a mark, and then relates it to a new transcription.
- **Select to Mark:** from the text editor, select part of the text and relate it to a mark generated on the fly. It gives the possibility to choose between a linear, rectangular, or polygonal mark.
- **Alternative Transcription:** In this strategy, the user proposes a new transcription for an existing mark-transcription relationship, as a way of correcting its content.
- **Semantic Transcription:** Maintains the same principle as the “mark to load” method, only that the mark is related to a Semantic Transcription, using vocabularies and ontologies (detailed below).

Alternative Transcription It is possible that during the transcription process, multiple interpretations of a document fragment may arise, leading to debate. Handling these different alternatives provides a higher level of participation and encourages collaboration to improve transcription. In this way, the “Alternative Transcription” strategy allows collaborators to propose other interpretations to the one initially generated by another user. For a particular tag, Transcriptor allows listing all the transcriptions suggested by users and allows creating a new alternative transcription to the current one.

3.3 Transcription Competition

To carry out the selection process of the primary transcript, the platform uses the voting method based on likes and, based on this value, selects the primary transcript as the one with the most positive votes.

Marks on the document



Transcription template

```

1 <span class="contribution-mark-13">{{ contributionMark13 }}</span>
2 <span class="contribution-mark-14">{{ contributionMark14 }}</span>
3 <span class="contribution-mark-15">{{ contributionMark15 }}</span>
4 <span class="contribution-mark-16">{{ contributionMark16 }}</span>
5 <span class="contribution-mark-17">{{ contributionMark17 }}</span>
6 <span class="contribution-mark-18">{{ contributionMark18 }}</span>

```

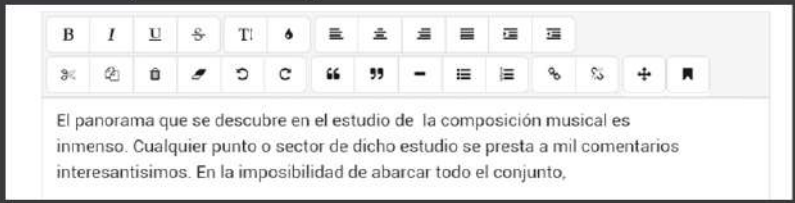
Template with dynamic replacements

```

1 <span class="contribution-mark-13">El panorama que se descubre en el estudio de</span>
2 <span class="contribution-mark-14">la composición musical es inmenso.</span>
3 <span class="contribution-mark-15">cualquier punto</span>
4 <span class="contribution-mark-16">o sector de dicho estudio se presta a mil co</span>
5 <span class="contribution-mark-17">mentarios interesantísimos. En la imposibilidad de</span>
6 <span class="contribution-mark-18">abarcarse todo el conjunto,</span>

```

Generated text



El panorama que se descubre en el estudio de la composición musical es inmenso. Cualquier punto o sector de dicho estudio se presta a mil comentarios interesantísimos. En la imposibilidad de abarcar todo el conjunto,

Fig. 4. Template transcription.

3.4 Version Control

Each time a change is made to the transcription of a page, Transcriptor takes a snapshot of the current state of the generated text, persisting a new version of the transcription. This feature allows the platform to maintain, as changes are made to the text, a list of changes to the document that can be traced back to who made them and when. Comparisons are made using the diff tool and can be visualized as shown in Figure 5.



Fig. 5. Transcription diff.

3.5 Semantic Layers Definition

Semantic marks are marks that, instead of describing only text, can describe a real-world object, such as a person, an organization, a book, an event, or a place, among many others, to represent those elements that are not easy to express in a transcription.

Because of its flexibility and breadth of defined concepts, the schema.org vocabulary was used for this example, which provides a language broad enough to represent the properties of real-world objects. Schema.org [5] offers a hierarchy of data types, which starts from the type “Thing” and has as children other classes such as “Action,” “Place,” “Person,” “CreativeWorks,” among others, these, in turn, have sub-classes, completing a hierarchy.

3.6 Gamification

These strategies were developed using the Metagame framework, which has a Rest API that allows registering different activities performed on the platform

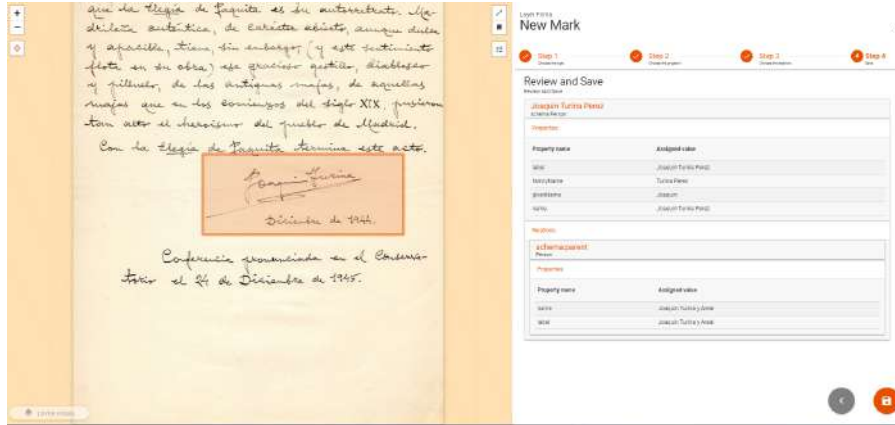


Fig. 6. Semantic annotation: the picture of a signature is annotated as an instance of schema:Person

and distinguished into "transcriptor types". As the user makes contributions on Transcriptor, they accumulate experience, which is reflected in medals and promotion of ranks. Obtaining these medals translates into promotion through the hierarchy of ranks.

3.7 Technologies Used and Implementation

Transcriptor Architecture

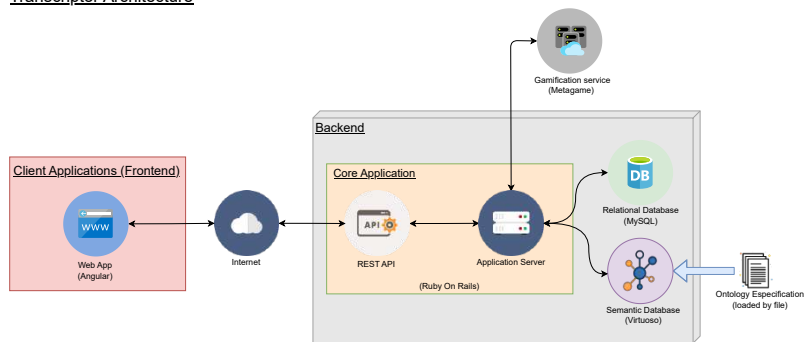


Fig. 7. Transcriptor architecture

Transcriptor is developed following a client-server architecture as is shown in Figure 7. The front-end⁷ and back-end⁸ parts could be downloaded from the GIT repository. Angular, Materialize, Ngx-Materialize, Leaflet, Leaflet Draw, RxJS, Ngx-Translate, Ruby On Rails, and RMagick were used in its implementation.

Particularly in gamification aspects, Metagame was a metagame system, which through the registration of activities in the application, establishes scores and assigns medals and ranks to the users who perform them. The semantic aspects were stored in a Virtuoso Universal Server, and the application code uses Ruby RDF.

Transcriptor uses MySQL as a relational database engine due to its flexibility and scalability, wide availability in operating systems, and ease of installation and use from Ruby through the `mysql2` gem.

4 Evaluation

An evaluation was conducted to assess the platform’s usability and user experience (UX). Two types of evaluations were conducted: a usability evaluation with end users, and a cognitive walkthrough evaluation.

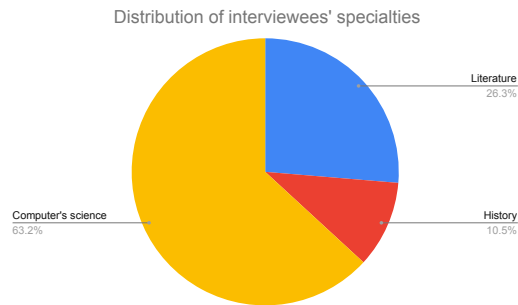


Fig. 8. Distribution of interviewees specialties

4.1 Usability Evaluation with End Users

This test seeks to obtain information about the level of ease of use of the platform to evaluate its ability to fulfill its purpose. For this, a series of activities were proposed, which are focused on the success/failure of the user to perform them and measuring the time required to execute them. To do so, the following questions will be answered:

⁷ <https://github.com/cientopolis/transcriptor-web> accessed on 7th february 2023

⁸ <https://github.com/cientopolis/transcriptor-backend> accessed on 7th february 2023

1. Is the platform easy to use?
2. Is it complex to learn how to use the platform?
3. Could Transcriptor be considered an innovative solution in the field of transcription?
4. Does the site and activities motivate the user to continue participating?

Additionally, a set of questions are included to capture the user's experience and level of compliance with the system's tasks and work environment. Thus, the evaluation was organized into three parts:

- Evaluation by Task: a UEQ [9, 8] test reduced by activity performed by the user, consisting of a questionnaire of five questions, which consist of pairs of opposing adjectives along with levels that mark a scale, where whoever completes the questionnaire must select which scale best represents their experience with the platform, each question refers to properties of transparency, attractiveness, and novelty.
- Overview of the platform: Short open-ended development questions on tested functionalities (5 questions).
- SUS Test: Complete in its positive version [7], which consists of ten items, with statements on a scale of 1 to 5. The participant must indicate how much they agree with such a statement, marking with one completely disagreeing and five completely agreeing.

4.2 Metrics

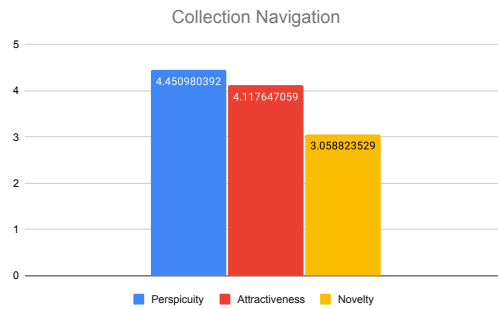


Fig. 9. Collection navigation EQ

The following metrics were considered to answer the questions listed in the previous section:

- Question 1 (Is the platform easy to use?) and question 2 (Is it complex to learn how to use the platform?)

- The SUS score will be compared with the scale defined by Bangor, Kortum, and Miller (2008) [1]. That scale specifies that, with a SUS score between 0 and 20.3, the level of usability would be the “worst imaginable”, between 20.3 and 37.7 would be “bad”, and between 37.7 and 50.9 is “regular”, between 50.9 and 71.4 is “acceptable”, between 71.4 and 81.5 “good”, between 81.5 and 90.1 “excellent” and between 90.1 and 100 the usability would be the “best imaginable”. To meet the questions posed, the SUS score is expected to be above 70 points, which would reach the “acceptable” level. In addition to these points, we will use the UEQ coefficients of “transparency” that allow us to determine how simple the platform is to communicate and achieve the objectives set out; if the average of these is above 3 points could be considered as the system is transparent and confirm the questions initially presented.
- Question 3 (Could Transcriptor be considered an innovative solution in the field of transcription?).
 - To confirm this question, the UEQ indicator of “novelty” will be used, denoting the level of innovation presented by the system’s functionalities. Comparing the average calculated between the system’s main characteristics, it is expected that this scale exceeds 3 points.
- Question 4 (Does the site and activities motivate the user to continue participating?).
 - In the same way that for question 3, the “novelty” indicator was used to establish compliance, the UEQ scale of “attraction” will be used for this approach. This average allows us to develop the level of interest that the system’s functionalities arouse in the users. Obtaining an average score higher than three among all the features would corroborate this point.

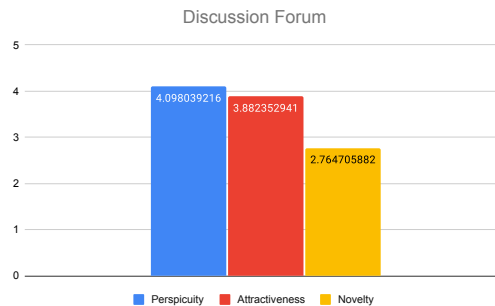


Fig. 10. Discussion Forum EQ

Evaluation Setup An open call was launched for users interested in testing the platform. The proposal was to conduct one-hour individual virtual interviews, which were recorded to keep a record of the participants’ reactions during the evaluation. The interviews followed the explain-do modality, so a moment was given to explain and perform the activity. After each interview, the developed questionnaire was sent to the participants by e-mail, which they completed anonymously through Google Forms.

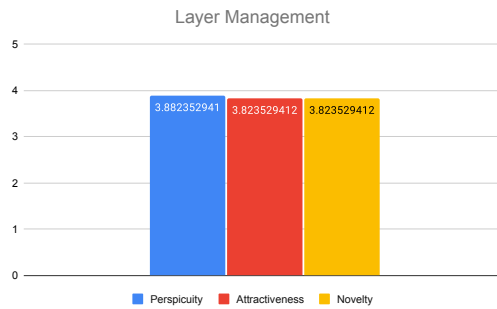


Fig. 11. Layer management EQ

Nineteen interviews were conducted with the participation of people of different ages, levels of study, and professionals, including computer specialists, historians, philologists, and researchers in the field of literature (see Figure 8).

Results The 19 questionnaires submitted by the test participants were tabulated and analyzed by separating each test section (UEQ test, open-ended questions, and SUS test).

For the UEQ test, the values were totaled by functionality, and an average was calculated for each of the scales corresponding to the pairs of properties used. In this way, a value between 1 and 5 was obtained, which we can define as the level of compliance with this scale, where 1 represents non-compliance with the scale, and 5 illustrates total compliance with it. Based on these results, bar graphs compared the rankings by functionality. Figure 9 describes the values for Collection Navigation task, Figure 10 shows the values for Discussion forum task, Figure 11 the values for Layer management task, Figure 12 shows the Semantic annotation management task.

Analyzing the graphs in Figure 13, it is possible to notice relationships between scales. For example, between that novelty and transparency, while in the most common functions of the platform (present in other systems, such as “Discussion forums”), the novelty index is lower than that of transparency. On the other hand, in the functions more linked to business logic, such as “Document

transcription mechanism,” these scales are more equal, with a lower transparency index but a higher novelty. This allows us to deduce that the initially mentioned features represented less innovation and are easier to learn to use; on the other hand, the own functionalities have been more innovative but suppose a more significant challenge to the user. Without experience from previous systems, the learning curve would be steeper.

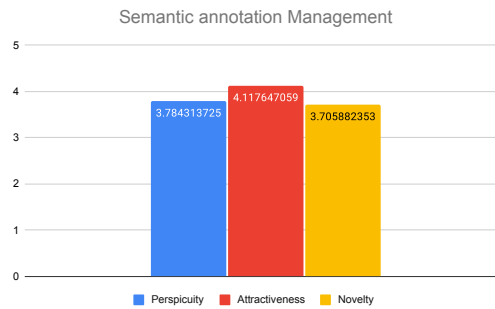


Fig. 12. Semantic annotation management EQ

The attractiveness index generally falls between transparency and novelty, indicating a dependency between these scales and reinforcing the idea that “if it is easy to use and novel in a certain aspect, then it is attractive”. Based on the above metrics, Transcriptor is considered an innovative solution in its field with an average UEQ coefficient of the novelty of 3.39 which exceeds the acceptance value for question 3. Similarly, for question 4, given that the average UEQ coefficient of attractiveness was 3.96 points, the acceptance threshold was exceeded, confirming an acceptable level of motivation for collaboration between users.

As for the SUS test, it obtained the average total value per question as shown in Figure 14. The final average SUS score is 71.6. Bangor, Kortum, and Miller (2008) [1] define in their study a scale based on seven adjectives that correspond directly to intervals in the scoring metric used in SUS. Using this scale of adjectives, the score obtained was compared by placing the platform’s user experience at a “Good” level, which assumes that the user experience is within the 70th percentile of the sites analyzed in the study above. This means that the user experience is remarkable and that the user will have no significant problems interpreting the site’s content.

Considering the results obtained, it was concluded that the platform is simple to use and easy to learn. This is justified since the average SUS score exceeded the acceptance threshold set at 70 points. The same is true for the UEQ coefficient for transparency, which was 4.11 (see figure 13), which exceeds the acceptance value set at 3.

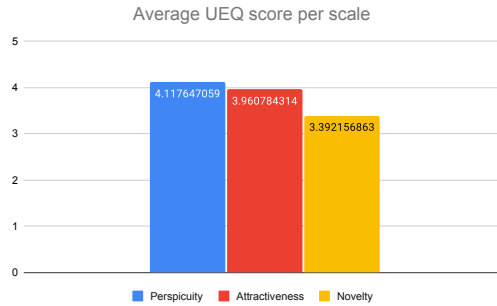


Fig. 13. Average UEQ scores.

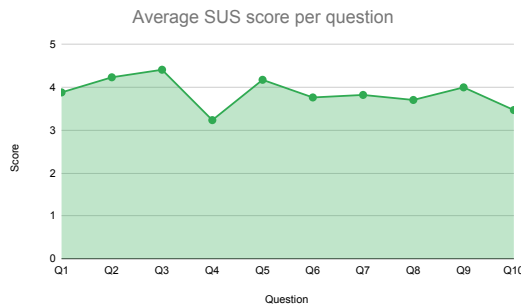


Fig. 14. Average SUS score per question

For the open-ended questions, the participants agreed that they considered the purpose of the transcriber to be useful, emphasizing the ease of use, the collaboration between users, the novelty of the information curation system, and the possibility of describing elements semantically. As a unanimous response, respondents indicated that the platform enhances collaboration, mentioning that this quality is due to the tool’s framework for a repetitive and complex process, such as the transcription of manuscripts. Ninety percent of the interviewees responded that they had no problems using the platform. The remaining 10 % reported minor bugs that did not prevent them from using the system, such as forms that did not clear and problems viewing certain lines of the manuscripts when selecting them with the “line selection” tool.

Teniendo en cuenta las preguntas “Do you find the transcription mechanism intuitive to use? Would you change anything?”, users agreed that the mechanism is attractive and easy to use. Still, they generally needed a guide to use it for the first time, especially the semantic transcription functionality. In this context,

most proposed establishing an initial walkthrough mechanism to introduce them to the tools and workspace available when making a transcription.

4.3 Cognitive Walkthrough

The objective of this evaluation is to obtain information about the level of ease of use and learning of the platform through the usability inspection method called Cognitive Walkthrough, which consists of a group of experts exploring the user interface, performing previously defined actions to evaluate the ease of learning of a platform design, in this case, the ease of use of Transcriptor. For the development of the test, the model defined by Granollers [4] was taken into account, which defines three steps for the creation of a cognitive path:

Data definition: The characteristics of the users are identified and documented. The description of the users will include the specific accumulated experience and acquired knowledge as determining factors for the verification of the “cognitive” factor during the tour. Then the prototype to be used in the evaluation is described. This prototype does not need to be highly detailed. Next, specific tasks to be performed are listed. Finally, for each task, implement a set of activities to carry them out.

Walk through the actions: The evaluators perform each task determined above by following the specified steps and using the detailed prototype. In this process, the evaluator will use information from the user’s cognitive factors (experience and acquired knowledge) to check if the interface is suitable for the user. This review must be thorough for all the actions specified to accomplish the task.

To do this, the evaluator at each action will critique the system by answering the following questions: Will the users do the right thing at this step? If they do, will they know if they did the right thing and are progressing toward the goal?

Document the results: The evaluator will note for each action the system responses and annotations. The output is a Usability Problem Report Sheet.

The following sections will describe the definition of each of these tasks applied in Transcriptor.

Data definition: An analysis of the user profile of the platform was carried out. Taking into account its objective and functionalities, it can be concluded that, although Transcriptor is available to all users who want to contribute to the transcription of manuscripts, most of the profiles may be linked to digital humanities, philology, or history activities. These are fields that are familiar with the loading of forms, the handling and loading of digital documentation, and the handling of semantic data about concepts.

This user archetype was used to establish the series of tasks to be analyzed during the tour. Each task was further subdivided into actions, which are the steps to complete the objective of the task. These are listed below:

- Browse collections, projects, and pages

- Adding a new transcript layer
- Perform a basic transcription
- Perform a transcription with markup
- Perform a Semantic transcription
- Post a comment in the discussion forum
- Vote for a transcription
- Export the data of a collection

Walk through the actions: A server was set up with the platform on which several documents were uploaded for transcription. In this context, a meeting between computer specialists was organized, and the system was run together, performing the tasks defined in the section 4.3. For each action that composes the task, the pair of questions given by the model were answered. These are: Will the user do the right thing in this step? If he does, will he know if he did the right thing and if he is progressing toward the goal? As the data for each response was obtained, it was recorded for later analysis.

Results documentation: Based on the data obtained, a table with the answers given was created to simplify the analysis of the responses. Furthermore, this analysis aims to detect potential usability problems in the covered functions. Based on this study, a report was prepared with the usability problems found, classifying them by severity and proposing a possible solution. These problems are detailed in Figure 15.

Task	Action	Issue	Severity	Solution alternative
Transcription with markup	Click on the first line of the image begins and click on the end of the line.	Depending on the text it is complex for the user to correctly select the beginning and/or end of the line and the mark is drawn in an incorrect section of the manuscript.	1 (slow)	1) Add a reference grid over the manuscript to act as a marking guide for the lines. 2) Automatically recognize the lines through image processing so that the user can click on them to load the markup.
Basic transcription	Save changes	If the user forgets to click on save changes to the transcript when exiting the transcript screen, the progress made so far will be lost.	3 (middle)	1) Add a confirmation when exiting the transcription screen if you have unsaved changes. 2) Notify the user that they have unsaved changes. 3) Alternatively, implement an automatic save of changes every certain amount of time.

Fig. 15. Cognitive walkthrough results

5 Conclusion and Further Work

Transcriptor provides a simple workflow, that enable to collaborate in a project all those persons interested (of any educational level), view, vote and comment other user's transcriptions. This presents an efficient solution to problems in the automated recognition of handwritten text, promoting initiatives such as crowdsourcing and citizen science.

On the other hand, the loss of positional and graphic information that affects the transcription of manuscripts is resolved by the platform through the use of a marking system. This system allows the creation of positional marks and semantic labels that help to preserve the positional context of the original corpus and describe graphic elements in the physical document.

Based on the objectives initially set as well as the evaluations carried out on the platform, it is considered that Transcriptor is an effective solution to the current problems of manuscript transcription. Transcriptor generates a workspace in which both specialists and amateurs can collaborate on real transcription projects promoted by organizations in the field.

The generated interaction between participants promotes the exchange of knowledge, benefiting both those responsible for the project and each collaborator, enriching themselves with the experience and stimulating continuous learning in collaboration and transcribed content, providing the community with easy-to-read, search and use material without losing the essence and details that exist in an original manuscript.

As further work, several extensions to the platform are planned. It can be mentioned the inclusion of automatic assisted transcription tools by recognition methods such as OCR or progress recognition. The semantic aspects could be extended by an easy manner to import other ontologies and vocabularies, even when the back-end is ready for that. The gamification model can be improved by the use of adaptive gamification which proposes a particular gamification activity according to each user profile and behavior in the Transcriptor use.

References

1. Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies* **4**(3), 114–123 (2009)
2. Bazillon, T., Estève, Y., Luzzati, D.: Manual vs assisted transcription of prepared and spontaneous speech. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco (May 2008), http://www.lrec-conf.org/proceedings/lrec2008/pdf/277_paper.pdf
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific american* **284**(5), 34–43 (2001), publisher: JSTOR
4. Granollers, T., Lorés, J.: Incorporation of users in the evaluation of usability by cognitive walkthrough. In: *HCI related papers of Interacción 2004*. pp. 243–255. Springer (2006)
5. Guha, R.V., Brickley, D., Macbeth, S.: Schema.org: evolution of structured data on the web. *Communications of the ACM* **59**(2), 44–51 (2016)

6. Huitfeldt, C., Sperberg-McQueen, C.: What is transcription? *Literary and Linguistic Computing* **23**, 295–310 (09 2008). <https://doi.org/10.1093/lc/fqn013>
7. Kortum, P., Acemyan, C.Z., Oswald, F.L.: Is it time to go positive? assessing the positively worded system usability scale (sus). *Human factors* **63**(6), 987–998 (2021)
8. Laugwitz, B., Held, T., Schrepp, M.: Construction and Evaluation of a User Experience Questionnaire. In: Holzinger, A. (ed.) *HCI and Usability for Education and Work*. pp. 63–76. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
9. Laugwitz, B., Schrepp, M., Held, T.: Konstruktion eines fragebogens zur messung der user experience von softwareprodukten. In: *Mensch & Computer*. pp. 125–134 (2006)
10. Memon, J., Sami, M., Khan, R.A., Uddin, M.: Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access* **8**, 142642–142668 (2020). <https://doi.org/10.1109/ACCESS.2020.3012542>
11. Mori, S., Nishida, H., Yamada, H.: *Optical character recognition*. John Wiley & Sons, Inc. (1999)
12. Moyle, M., Tonra, J., Wallace, V.: Manuscript transcription by crowd-sourcing: Transcribe bentham. *Liber Quarterly* **20**, 3 (01 2010). <https://doi.org/10.18352/lq.7999>
13. Ortiz Baco, J., Guzman, A., Palacios, A.A.: Fromthepage collection owner user study report. *Enabling and Reusing Multilingual Citizen Contributions in the Archival Record-NEH Grant Documentation* (2020)
14. del Rio, M.G., Hernández, N., De León, R.: Manos en la masa (textual): introducción al editatón colaborativo con fromthepage. In: *III Congreso Internacional de la Asociación de Humanidades Digitales (AAHD)*. *La cultura de los datos* (2018)
15. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. *IEEE intelligent systems* **21**(3), 96–101 (2006), publisher: IEEE
16. Simpson, R., Page, K.R., De Roure, D.: Zooniverse: observing the world’s largest citizen science platform. In: *Proceedings of the 23rd international conference on world wide web*. pp. 1049–1054 (2014)
17. Vershbow, B.: *Scribe: Turning text into structured information through the power of the crowd* (2015)

Apéndice B

Propuesta Evaluación de usuarios

Propuesta de diseño para evaluación de Transcriptor

Objetivo

Obtener información acerca del nivel de facilidad de uso de la plataforma para evaluar la capacidad que ésta tiene para cumplir su propósito. Para esto, se propondrán una serie de actividades las cuales están enfocadas en el éxito/fracaso del usuario para realizarlas y cuánto tiempo le conlleva ejecutarla.

Finalmente se presentará una serie de preguntas que tienen como objetivo capturar la experiencia y el nivel de conformidad del usuario frente a las tareas y al entorno de trabajo del sistema.

Método

Organizar un taller o reunión donde participen transcriptores para que utilicen Transcriptor, con la finalidad de obtener un feedback de la solución propuesta en esta tesina.

Tareas para la Organización

1. Instalar Transcriptor en un servidor y configurarlo con acceso público.
2. Plantear una serie de actividades, con las características principales de Transcriptor, las cuales pueden ser:
 - Navegación de colecciones, proyectos y páginas.
 - Transcripciones y marcas en los documentos.
 - Sistema de votación de transcripciones.
 - Hacer uso del foro de discusión.
 - Utilización de capas.
 - Manejo de marcas semánticas y exportación.
3. Se realizarán entrevistas individuales a 15 personas las cuales podrán acceder y hacer uso de transcriptor para realizar actividades propuestas.
4. Durante cada entrevista se mostrará un video por actividad y se medirá el tiempo de realización para detectar deficiencias / dificultades en la utilización del sitio.
5. Asignar un cuestionario a cada uno de los transcriptores, con preguntas para hacer la evaluación pertinente, a continuación proponemos un modelo de las posibles preguntas para dicha encuesta..

Acerca del cuestionario

Para definir la evaluación, se planteó usar tanto test de usabilidad (del inglés, System Usability Scale, SUS) junto con Cuestionario de experiencia de usuario (User Experience Questionnaire, UEQ) y preguntas abiertas.

El test de SUS lo utilizaremos en la sección de *conclusiones*. Consiste en un cuestionario de 5 preguntas por actividad, donde se presentan afirmaciones que tendrán la escala de 1 a 5, indicando que tan de acuerdo se siente el participante con el proyecto, siendo 1 completamente en desacuerdo y 5 totalmente de acuerdo.

El test UEQ es utilizado en la sección *evaluación por tarea*, la cual consiste en colocar pares de adjetivos opuestos junto a niveles que marcan una escala, donde quien completa el cuestionario podrá seleccionar qué escala representa mejor su experiencia con la plataforma.

Transcriptor: Evaluación de usabilidad

Objetivo

Obtener información acerca del nivel de facilidad de uso de la plataforma para evaluar la capacidad que ésta tiene para cumplir su propósito. Para esto, se propondrán una serie de actividades las cuales están enfocadas en el éxito/fracaso del usuario para realizarlas y cuánto tiempo le conlleva ejecutarla.

Finalmente se presentará una serie de preguntas que tienen como objetivo capturar la experiencia y el nivel de conformidad del usuario frente a las tareas y al entorno de trabajo del sistema.

Tareas a realizar

Para completar esta evaluación es necesario que realice las siguientes tareas dentro de la plataforma, teniendo en cuenta los pasos a realizar para lograr el objetivo y siendo crítico sobre cualquier aspecto durante su realización. Antes de intentar la actividad mirar el tutorial de cada una.

Nota: Intente realizar las actividades de manera individual, pero no dude en consultar en el caso que se presenten dificultades.

Actividades

N°	Actividad	Completado
1	Listar las paginas del proyecto "Viaje al Río de la Plata" de la colección "Schmidel".	
2	Ingresar a la pagina 27, marcar y transcribir la línea "crónicas é historias escritas antes ó después sobre".	
3	Proponer una nueva transcripción para la línea "actos formaron los soldados colonizadores que".	
4	Abrir la ventana de votación y votar por la transcripción "ción inicial del Río de la Plata por la raza europea."	
5	Agregar la capa "Personas", seleccione y cargue una marca semántica para el fragmento que menciona a "don Pedro de Mendoza" con los siguientes datos: <ul style="list-style-type: none">● Ontología: FOAF● Clase: Person● Propiedades Simples:<ul style="list-style-type: none">○ Name: Pedro○ Surname: de Mendoza○ Description: Militar de familia noble, caballero de Alcántara desde 1524, almirante2 y conquistador español, nombrado por el emperador Carlos V como el primer adelantado del Río de la Plata y como gobernador de la Nueva Andalucía	

6	Ingresar al detalle de la nueva marca y visualizar los datos cargados.	
7	A través del buscador semántico realizar una búsqueda de la información que cargamos ingresando el texto "Pedro de Mendoza".	

Panorama general de la plataforma

1. ¿Encuentra útil la finalidad de transcriptor?
2. ¿Piensa que Transcriptor potencia la colaboración entre usuarios?
3. ¿Encontró dificultades al momento de generar transcripciones?
4. ¿Encuentra atractivo el mecanismo de transcripción? ¿Cambiarías algo?

Evaluación por tarea

Marque un círculo para cada par de propiedades, aunque piense que no son aplicables o que hay propiedades parecidas o prácticamente iguales. Por favor, decida espontáneamente. No piense demasiado en su opinión y asegúrese que expresa su sensación inicial.






- **Navegación de colecciones.**

No entendible	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Entendible
Fácil de aprender	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Difícil de aprender
Complicado	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Fácil
Cómodo	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Incómodo
Convencional	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Innovador






- **Transcripciones y marcas en los documentos.**

No entendible	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Entendible
Fácil de aprender	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Difícil de aprender
Complicado	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Fácil
Obstrutivo	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Impulsor de apoyo
Convencional	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Innovador






- **Sistema de votación de transcripciones.**

No entendible		Entendible
Fácil de aprender		Difícil de aprender
Complicado		Fácil
Obstructivo		Impulsor de apoyo
Convencional		Innovador






- **Foros de discusión**

No entendible		Entendible
Fácil de aprender		Difícil de aprender
Complicado		Fácil
Obstructivo		Impulsor de apoyo
Convencional		Innovador

- **Utilización de capas**

No entendible		Entendible
Fácil de aprender		Difícil de aprender
Complicado		Fácil
Obstructivo		Impulsor de apoyo
Convencional		Innovador

- **Manejo de marcas semánticas y exportación.**

No entendible		Entendible
Fácil de aprender		Difícil de aprender
Complicado		Fácil
Obstructivo		Impulsor de apoyo
Convencional		Innovador

Conclusiones

Califique cada afirmación con un número de 1 a 5, según su nivel de concordancia con cada una, siendo 1 completamente en desacuerdo y 5 completamente de acuerdo.

Afirmación	Puntaje				
Creo que me gustaría utilizar frecuentemente este sitio web.	1	2	3	4	5
Encontré el sitio web sencillo.	1	2	3	4	5
Pienso que el sitio web es fácil de usar.	1	2	3	4	5
Pienso que podré utilizar este sitio web sin el apoyo de personal técnico.	1	2	3	4	5
Encontré que varias de las funciones en el sitio web estaban bien integradas.	1	2	3	4	5
Pensé que había demasiada consistencia en el sitio web.	1	2	3	4	5
Me imagino que la mayoría de las personas podrían aprender a usar este sitio web muy rápido.	1	2	3	4	5
Encontré el sitio web muy intuitivo.	1	2	3	4	5
Me sentí muy confiado (seguro) al utilizar el sitio web.	1	2	3	4	5
Pude utilizar el sitio web sin tener que aprender nada nuevo.	1	2	3	4	5

Formulario de entrevistador

Entrevista N°:

Nombre y apellido del tester:

Instrucciones

Complete las tablas según el desempeño del entrevistado, cronometrando la actividad y observando los errores y/o dificultades que se le presenten al usuario durante el proceso.

Actividades

N°	Descripción	Tiempo
1	Listar las páginas del proyecto "Viaje al Rio de la Plata" de la colección "Schmidel".	
Observaciones		

N°	Descripción	Tiempo
2	Ingresar a la pagina 2, marcar y transcribir la línea "una linea".	
Observaciones		

N°	Descripción	Tiempo
3	Proponer una nueva transcripción para la línea "otra linea".	
Observaciones		

N°	Descripción	Tiempo
4	Abrir la ventana de votación y votar por la transcripción ""	
Observaciones		

N°	Descripción	Tiempo
5	<p>Agregar la capa "Firmas", seleccione y cargue una marca semántica con los siguientes datos:</p> <ul style="list-style-type: none"> ● Ontología: FOAF ● Clase: Person ● Propiedades Simples: <ul style="list-style-type: none"> ○ Name: Juan ○ Surname: Perez 	
Observaciones		

N°	Descripción	Tiempo
6	Ingresar al detalle de la nueva marca y visualizar los datos cargados.	
Observaciones		

N°	Descripción	Tiempo
7	A través del buscador semántico realizar una búsqueda de la información que cargamos ingresando el texto "Juan Perez".	
Observaciones		

Apéndice C

Propuesta Recorrido cognitivo

Transcriptor: Evaluación de la usabilidad mediante recorrido cognitivo

Objetivo

El objetivo de esta evaluación es obtener información acerca del nivel de facilidad del uso y aprendizaje de la plataforma mediante el método de inspección de usabilidad denominado recorrido cognitivo (o Cognitive Walkthrough) el cual consiste en que un grupo de expertos exploren la interfaz de usuarios, realizando acciones previamente definidas para evaluar la facilidad de aprendizaje de un diseño de la plataforma, en este caso, la facilidad de uso de Transcriptor.

Definición del proceso

Los pasos necesarios para la realización del método se estructuran en el documento de la evaluación y son los siguientes:

1. Definición de los datos necesarios para el recorrido:

- Se identifican y documentan las características de los usuarios ¿Quiénes serán los usuarios del sistema? La descripción de los usuarios incluirá la experiencia específica acumulada y el conocimiento adquirido como factores determinantes para la comprobación del factor “cognitivo” durante el recorrido.
- Se describe también el prototipo a utilizar para la evaluación, que no es preciso que sea ni completo ni tener un gran nivel de detalle.
- Se enumeran las tareas concretas a desarrollar.
- Para cada tarea se implementa por escrito, la lista íntegra de las acciones necesarias para completar la tarea con el prototipo descrito. Esta lista consta de una serie repetitiva de pares de acciones (del usuario) y respuestas (del sistema).

2. Recorrer las acciones:

Los evaluadores realizan cada una de las tareas determinadas anteriormente siguiendo los pasos especificados y utilizando el prototipo detallado. En este proceso, el evaluador utilizará la información del factor cognitivo (experiencia y conocimiento adquirido) de los usuarios para comprobar si la interfaz es adecuada para el mismo. Esta revisión ha de ser minuciosa para todas las acciones especificadas para la consecución de la tarea.

Para ello, el evaluador en cada acción criticará el sistema respondiendo a las siguientes preguntas:

- a. ¿El usuario hará lo correcto en este paso?
- b. Si lo hace, ¿sabrá si hizo lo correcto y está progresando hacia la meta?

3. Documentar los resultados

- El evaluador anotará para cada acción las respuestas del sistema y sus anotaciones.
- El documento incluirá un anexo especial, conocido como Reporte de problemas de usabilidad (Usability Problem Report Sheet) detallando los aspectos negativos de la evaluación relacionándolos con un grado de severidad que permita distinguir aquellos errores más perjudiciales de los que no lo son tanto.

Definición de los datos necesarios para el recorrido cognitivo.

Características de los usuarios:

Si bien Transcriptor está disponible para todos los usuarios que quieran ayudar a transcribir manuscritos, podemos asumir que la mayor parte de los usuarios estarán relacionados con actividades de humanidades digitales o bibliotecarias. Estos usuarios están familiarizados con la carga de formularios y el manejo y carga de documentación digital.

El prototipo

El recorrido se realizará en una plataforma de pruebas montada en un servidor público

(link de la plataforma)

Tareas y acciones a desarrollar

- Navegación de colecciones, proyectos y páginas
 1. Hacer click en el menú “Colecciones”.
 2. Buscar y hacer click en la colección llamada ‘Schmidel’.
 3. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
 4. Buscar y hacer click en la página 1.
- Utilización de capas
 1. Hacer click en el menú “Colecciones”
 2. Buscar y hacer click en la colección llamada ‘Schmidel’.
 3. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
 4. Buscar y hacer click en la página 1.
 5. Hacer click en “Agregar capa”.
 6. Elegir un nombre de su interés y hacer click en guardar.

- Transcripciones, marcas y marcas semánticas en los documentos

1. Transcripción Semántica

- a. Hacer click en el menú “Colecciones”.
- b. Buscar y hacer click en la colección llamada ‘Schmidel’.
- c. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
- d. Buscar y hacer click en la página 1.
- e. Hacer click en “Agregar capa”
- f. Escribir como nombre de capa “Firma” y hacer click en guardar.
- g. Seleccionar la capa “Firma”.
- h. Buscar en los controles de la imagen la opción “agregar marca”.
- i. Marcar el área donde se encuentra la firma.
- j. Elegir la ontología FOAF en el panel derecho de la pantalla y click en el botón siguiente,
- k. Elegir el datatype Person y hacer click en el botón siguiente.
- l. Seleccionar las data properties Name & Surname y hacer click en el botón siguiente.
- m. Click siguiente sin agregar ninguna relación
- n. Llenar los campos Name y Surname con los datos: Juan y Perez respectivamente y hacer click en el botón siguiente.
- o. Hacer click en guardar.

2. Transcripción básica

- a. Hacer click en el menú “Colecciones”.
- b. Buscar y hacer click en la colección llamada ‘Schmidel’.
- c. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
- d. Buscar y hacer click en la página 1.
- e. En la entrada derecha de la pantalla, escribir algo de lo que lea en la imagen del manuscrito.
- f. Al texto ingresado, darle formato al primer párrafo con el panel superior, buscar la opción “negrita”.
- g. Hacer click en guardar.

3. Transcripción con marca

- a. Hacer click en el menú “Colecciones”.
- b. Buscar y hacer click en la colección llamada ‘Schmidel’.
- c. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
- d. Buscar y hacer click en la página 1.
- e. Buscar en los controles de la imagen la opción “agregar marca”.
- f. Hacer click en donde comienza el primer renglón de la imagen y hacer click en el final del mismo.
- g. Escribir lo que se alcanza a leer en el área seleccionada.
- h. Revisar lo escrito y hacer click en guardar la transcripción.

- Hacer uso del foro de discusión
 1. Hacer click en el menú “Colecciones”.
 2. Buscar y hacer click en la colección llamada ‘Schmidel’.
 3. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
 4. Buscar y hacer click en la página 1.
 5. Hacer click en el alguna marca que aparece en la imagen del documento.
 6. Hacer click en el botón llamado “Foro”.
 7. Escribir un comentario y hacer click en botón guardar.

- Sistema de votación de transcripciones
 1. Hacer click en el menú “Colecciones”.
 2. Buscar y hacer click en la colección llamada ‘Schmidel’.
 3. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
 4. Buscar y hacer click en la página 1.
 5. Buscar y hacer click en una marca.
 6. Hacer click en el botón otras transcripciones.
 7. Buscar una transcripción y hacer click en el botón “Me gusta”.

- Manejo de marcas semánticas y exportación
 1. Hacer click en el menú “Colecciones”.
 2. Buscar y hacer click en la colección llamada ‘Schmidel’.
 3. Buscar y hacer click en el proyecto “Viaje al Río de La Plata”.
 4. Buscar y hacer click en la página 1.
 5. Hacer click en el menú “Colecciones”
 6. Buscar y hacer click en la colección llamada ‘Schmidel’
 7. Hacer click en el boton de configuracion
 8. Hacer click en el botón exportar colección