



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE SEGOVIA**

**Grado en Ingeniería Informática  
de Servicios y Aplicaciones**

---

**Annotator: una herramienta de anotación de  
textos asistida por aprendizaje automático**

---

**Alumno: Carlos Cubo Izquierdo**

**Tutores: Aníbal Bregón Bregón  
Jorge Silvestre Vilches**



# Annotator: una herramienta de anotación de textos asistida por aprendizaje automático

Carlos Cubo Izquierdo



*Solo aprende  
quien se equivoca.*

*Al final,  
todo esfuerzo da su fruto.*



# Agradecimientos

Quisiera agradecer a mis tutores Aníbal Bregón y Jorge Silvestre, así como al profesor Miguel Ángel Martínez, su gran labor docente y predisposición durante estos meses de proyecto en los que me han asesorado y guiado en todo el proceso.

Y, sobre todo, me gustaría dar inmensas gracias a mis padres por haber estado apoyándome siempre, acompañándome tanto en los buenos como en los no tan buenos momentos.







# Índice general

<b>Índice general</b>	<b>I</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de tablas</b>	<b>XI</b>
<b>Resumen</b>	<b>XVII</b>
<b>Abstract</b>	<b>XIX</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Contexto . . . . .	3
1.3. Visión y Alcance . . . . .	6
1.4. Objetivos del Proyecto . . . . .	8
1.5. Estructura del documento . . . . .	9
<b>2. Estado del Arte</b>	<b>13</b>
2.1. Prodigy . . . . .	13
2.2. Doccano . . . . .	15
2.3. Anafora . . . . .	16
2.4. Brat . . . . .	18
2.5. Tagtog . . . . .	19
2.6. Comparativa Herramientas de Anotación . . . . .	20
<b>3. Planificación y Presupuesto</b>	<b>23</b>
3.1. Metodología Utilizada . . . . .	23
3.1.1. Gestión del Esfuerzo . . . . .	27
3.2. Estimación . . . . .	28
3.3. Planificación Temporal . . . . .	32
3.4. Presupuesto . . . . .	37

<b>4. Fundamentos Teóricos</b>	<b>41</b>
4.1. Anotación . . . . .	41
4.2. Procesamiento del Lenguaje Natural . . . . .	42
4.3. Reconocimiento de Entidades Nombradas . . . . .	44
4.4. Reconocimiento Óptico de Caracteres . . . . .	47
4.5. Aplicación Web . . . . .	50
<b>5. Análisis</b>	<b>53</b>
5.1. Actores del Sistema . . . . .	53
5.2. Requisitos de Usuario . . . . .	53
5.3. Casos de Uso . . . . .	56
5.3.1. Especificación de Casos de Uso . . . . .	59
5.4. Requisitos Funcionales . . . . .	68
5.5. Requisitos No Funcionales . . . . .	79
5.6. Reglas de Negocio . . . . .	79
5.7. Requisitos de Información . . . . .	80
<b>6. Diseño</b>	<b>83</b>
6.1. Arquitectura Lógica . . . . .	83
6.2. Arquitectura Física . . . . .	85
6.3. Diagrama de Clases . . . . .	86
6.4. Diagramas de Secuencia . . . . .	88
6.5. Modelo Lógico de Datos . . . . .	97
6.5.1. Diccionario de Datos . . . . .	97
6.6. Diseño de Interfaz . . . . .	103
<b>7. Implementación</b>	<b>107</b>
7.1. Descripción Técnica . . . . .	107
7.2. Organización del Código . . . . .	111
7.3. Herramientas Empleadas . . . . .	118
7.4. Tecnologías Involucradas . . . . .	121
7.4.1. Anotación . . . . .	122
7.4.2. Reconocimiento de Entidades Nombradas . . . . .	122
7.4.3. Reconocimiento Óptico de Caracteres . . . . .	126
7.4.4. Aplicación Web . . . . .	129
<b>8. Pruebas</b>	<b>131</b>
<b>9. Manuales</b>	<b>139</b>
9.1. Manual de Instalación . . . . .	139
9.2. Manual de Usuario . . . . .	143

<b>10. Conclusiones y Trabajo Futuro</b>	<b>161</b>
10.1. Conclusiones . . . . .	161
10.2. Trabajo Futuro . . . . .	162
<b>Bibliografía</b>	<b>165</b>



# Índice de figuras

1.1. Proceso de entrenamiento de un modelo de aprendizaje . . . . .	3
1.2. Árbol de Características de Annotator . . . . .	7
2.1. Interfaz Herramienta Prodigy . . . . .	14
2.2. Interfaz Herramienta Doccano . . . . .	16
2.3. Interfaz Herramienta Anafora . . . . .	17
2.4. Interfaz Herramienta Brat . . . . .	18
2.5. Interfaz Herramienta Tagtog . . . . .	19
3.1. Marco de Trabajo Scrum . . . . .	27
3.2. Gestión del Esfuerzo . . . . .	28
3.3. Diagrama de Gantt Inicial . . . . .	34
3.4. Diagrama de Gantt Final . . . . .	36
4.1. Interfaz Gráfica de Anotación de Entidades . . . . .	42
4.2. Situación Procesamiento del Lenguaje Natural . . . . .	43
4.3. Proceso General OCR . . . . .	49
4.4. Flujo de Proceso HTTP . . . . .	50
5.1. Diagrama de Casos de Uso . . . . .	58
5.2. Diagrama Entidad-Relación . . . . .	81
6.1. Arquitectura Lógica de Alto Nivel . . . . .	84
6.2. Arquitectura Lógica Detallada . . . . .	85
6.3. Arquitectura Física . . . . .	86
6.4. Diagrama de Clases . . . . .	87
6.5. Diagrama de Secuencia relativo al Caso de Uso CU-01: Crear un proyecto de anotación. . . . .	88
6.6. Diagrama de Secuencia relativo al Caso de Uso CU-08: Editar los límites de una anotación. . . . .	89
6.7. Diagrama de Secuencia relativo al Caso de Uso CU-13: Editar el color de una categoría de entidad. . . . .	90
6.8. Diagrama de Secuencia relativo al Caso de Uso CU-22: Importar fichero PDF. . . . .	91

6.9. Diagrama de Secuencia relativo al Caso de Uso CU-24: Exportar documentos anotados. . . . .	92
6.10. Diagrama de Secuencia relativo al Caso de Uso CU-27: Importar modelo de aprendizaje preentrenado. . . . .	93
6.11. Diagrama de Secuencia relativo al Caso de Uso CU-29: Activar asistencia con patrones de coincidencia. . . . .	94
6.12. Diagrama de Secuencia relativo al Caso de Uso CU-30: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia. . . . .	95
6.13. Diagrama de Secuencia relativo al Caso de Uso CU-31: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia. . . . .	96
6.14. Modelo Lógico de Datos . . . . .	97
7.1. Árbol de Características Simplificado . . . . .	108
7.2. Pipeline de Procesamiento de Annotator . . . . .	110
7.3. Estructura Directorio Principal del Proyecto . . . . .	112
7.4. Estructura Directorio Core del Proyecto . . . . .	113
7.5. Fragmento Código urls.py . . . . .	113
7.6. Fragmento Código views.py . . . . .	114
7.7. Fragmento Código utils.py . . . . .	115
7.8. Fragmento Código models.py . . . . .	116
7.9. Fragmento Código herramienta.css . . . . .	116
7.10. Fragmento Código herramienta.js . . . . .	117
7.11. Estructura Directorio Proyectos del Proyecto . . . . .	117
7.12. Pipeline de Procesamiento de un texto . . . . .	123
7.13. Estructura red LSTM . . . . .	128
7.14. Patrón MVT de Django . . . . .	129
8.1. Esquema Pruebas de Caja Negra . . . . .	131
9.1. Aplicación Ubuntu Software . . . . .	140
9.2. Visual Studio Code en Ubuntu Software . . . . .	140
9.3. Página de Inicio Annotator . . . . .	142
9.4. Opción Crear Proyecto . . . . .	144
9.5. Opción Abrir Proyecto . . . . .	145
9.6. Opción Renombrar Proyecto . . . . .	145
9.7. Opción Eliminar Proyecto . . . . .	146
9.8. Opción Guardar Proyecto . . . . .	146
9.9. Opción Crear Expresión Regular . . . . .	147
9.10. Opción Eliminar Expresión Regular . . . . .	147
9.11. Opción Editar Expresión Regular . . . . .	147
9.12. Opción Asignar Expresión Regular a Categoría de Entidad . . . . .	148
9.13. Opción Exportar Fichero de Expresiones Regulares . . . . .	148
9.14. Opción Importar Fichero de Expresiones Regulares . . . . .	149
9.15. Opción Crear Anotación . . . . .	149

---

9.16. Opción Eliminar Anotación . . . . .	150
9.17. Opción Editar Límites de Anotación . . . . .	150
9.18. Opción Cambiar Categoría de Entidad de una Anotación . . . . .	151
9.19. Opción Crear Categoría de Entidad . . . . .	151
9.20. Opción Eliminar Categoría de Entidad . . . . .	151
9.21. Opción Editar Nombre de Categoría de Entidad . . . . .	151
9.22. Opción Editar Color de Categoría de Entidad . . . . .	152
9.23. Opción Seleccionar Categoría de Entidad . . . . .	152
9.24. Opción Importar Documentos de Texto . . . . .	153
9.25. Opción Importar Fichero PDF . . . . .	153
9.26. Opción Exportar Documentos Anotados . . . . .	154
9.27. Opción Importar Documentos Anotados . . . . .	154
9.28. Opción Navegar entre Documentos . . . . .	155
9.29. Opción Navegar entre Sentencias de Documento . . . . .	155
9.30. Opción Visualizar Estadísticas de Anotación . . . . .	156
9.31. Opción Establecer Opciones de Importación . . . . .	157
9.32. Opción Exportar Modelo de Aprendizaje Entrenado . . . . .	157
9.33. Opción Importar Modelo de Aprendizaje Preentrenado . . . . .	158
9.34. Opción Activar Asistencia con Patrones de Coincidencia . . . . .	158
9.35. Opción Activar Asistencia con Entrenamiento de Modelo de Aprendizaje y Pa- trones de Coincidencia . . . . .	159
9.36. Opción Activar Asistencia con Predicciones de Modelo de Aprendizaje Importado	160
9.37. Opción Visualizar Resultados de Entrenamiento . . . . .	160
10.1. Arquitectura Física propuesta para el despliegue de la herramienta en producción	163





# Índice de tablas

1.1. Objetivos del Proyecto . . . . .	8
1.2. Subobjetivo de OBJ-02 . . . . .	8
1.3. Criterios de aceptación para OBJ-01 . . . . .	9
1.4. Criterios de aceptación para OBJ-02 . . . . .	9
1.5. Criterios de aceptación para OBJ-02.01 . . . . .	10
1.6. Criterios de aceptación para OBJ-03 . . . . .	10
2.1. Matriz Comparativa Herramientas de Anotación . . . . .	21
3.1. Tareas asociadas a la Historia de Usuario HU-01 . . . . .	30
3.2. Tareas asociadas a la Historia de Usuario HU-02 . . . . .	30
3.3. Tareas asociadas a la Historia de Usuario HU-03 . . . . .	31
3.4. Tareas asociadas a la Historia de Usuario HU-04 . . . . .	31
3.5. Tareas asociadas a la Historia de Usuario HU-05 . . . . .	32
3.6. Planificación Temporal Inicial de los Sprints . . . . .	33
3.7. Planificación Temporal Final de los Sprints . . . . .	35
3.8. Prestaciones Ordenador Portátil de Desarrollo . . . . .	37
3.9. Costes de Hardware . . . . .	37
3.10. Coste de Software . . . . .	38
3.11. Coste de Recursos Humanos . . . . .	38
3.12. Otros Costes . . . . .	39
3.13. Presupuesto Inicial . . . . .	39
3.14. Coste de Recursos Humanos Actualizado . . . . .	40
3.15. Presupuesto Final . . . . .	40
5.1. Actor ACT-01 - Usuario . . . . .	53
5.2. Requisitos de Usuario . . . . .	54
5.3. Requisitos de Usuario . . . . .	55
5.4. Casos de Uso . . . . .	56
5.5. Casos de Uso . . . . .	57
5.6. Especificación Caso de Uso CU-01: Crear un proyecto de anotación. . . . .	59
5.7. Especificación Caso de Uso CU-02: Abrir un proyecto de anotación. . . . .	60
5.8. Especificación Caso de Uso CU-03: Renombrar un proyecto de anotación. . . . .	60
5.9. Especificación Caso de Uso CU-04: Eliminar un proyecto de anotación. . . . .	61

5.10. Especificación Caso de Uso CU-06: Añadir una anotación. . . . .	61
5.11. Especificación Caso de Uso CU-08: Editar los límites de una anotación. . . . .	62
5.12. Especificación Caso de Uso CU-11: Eliminar una categoría de entidad. . . . .	62
5.13. Especificación Caso de Uso CU-13: Editar el color de una categoría de entidad. . . . .	63
5.14. Especificación Caso de Uso CU-21: Importar documento de texto. . . . .	63
5.15. Especificación Caso de Uso CU-22: Importar fichero PDF. . . . .	64
5.16. Especificación Caso de Uso CU-23: Importar documentos anotados. . . . .	64
5.17. Especificación Caso de Uso CU-24: Exportar documentos anotados. . . . .	65
5.18. Especificación Caso de Uso CU-29: Activar asistencia con patrones de coincidencia. . . . .	65
5.19. Especificación Caso de Uso CU-30: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia. . . . .	66
5.20. Especificación Caso de Uso CU-31: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia. . . . .	67
5.21. Requisitos Funcionales relativos al Caso de Uso CU-01: Crear un proyecto de anotación. . . . .	68
5.22. Requisitos Funcionales relativos al Caso de Uso CU-02: Abrir un proyecto de anotación. . . . .	68
5.23. Requisitos Funcionales relativos al Caso de Uso CU-03: Renombrar un proyecto de anotación. . . . .	68
5.24. Requisitos Funcionales relativos al Caso de Uso CU-04: Eliminar un proyecto de anotación. . . . .	69
5.25. Requisitos Funcionales relativos al Caso de Uso CU-05: Guardar un proyecto de anotación. . . . .	69
5.26. Requisitos Funcionales relativos al Caso de Uso CU-06: Añadir una anotación. . . . .	69
5.27. Requisitos Funcionales relativos al Caso de Uso CU-07: Eliminar una anotación. . . . .	69
5.28. Requisitos Funcionales relativos al Caso de Uso CU-08: Editar los límites de una anotación. . . . .	70
5.29. Requisitos Funcionales relativos al Caso de Uso CU-09: Cambiar la categoría de una anotación. . . . .	70
5.30. Requisitos Funcionales relativos al Caso de Uso CU-10: Añadir una categoría de entidad. . . . .	70
5.31. Requisitos Funcionales relativos al Caso de Uso CU-11: Eliminar una categoría de entidad. . . . .	71
5.32. Requisitos Funcionales relativos al Caso de Uso CU-12: Editar el nombre de una categoría de entidad. . . . .	71
5.33. Requisitos Funcionales relativos al Caso de Uso CU-13: Editar el color de una categoría de entidad. . . . .	71
5.34. Requisitos Funcionales relativos al Caso de Uso CU-14: Seleccionar una categoría de entidad. . . . .	71
5.35. Requisitos Funcionales relativos al Caso de Uso CU-15: Crear una expresión regular. . . . .	72

5.36. Requisitos Funcionales relativos al Caso de Uso CU-16: Eliminar una expresión regular. . . . .	72
5.37. Requisitos Funcionales relativos al Caso de Uso CU-17: Editar una expresión regular. . . . .	72
5.38. Requisitos Funcionales relativos al Caso de Uso CU-18: Asociar una expresión regular a una categoría de entidad. . . . .	72
5.39. Requisitos Funcionales relativos al Caso de Uso CU-19: Exportar fichero con expresiones regulares. . . . .	73
5.40. Requisitos Funcionales relativos al Caso de Uso CU-20: Importar fichero con expresiones regulares. . . . .	73
5.41. Requisitos Funcionales relativos al Caso de Uso CU-21: Importar documento de texto. . . . .	73
5.42. Requisitos Funcionales relativos al Caso de Uso CU-22: Importar fichero PDF. . . . .	74
5.43. Requisitos Funcionales relativos al Caso de Uso CU-23: Importar documentos anotados. . . . .	74
5.44. Requisitos Funcionales relativos al Caso de Uso CU-24: Exportar documentos anotados. . . . .	74
5.45. Requisitos Funcionales relativos al Caso de Uso CU-25: Navegar entre los documentos. . . . .	75
5.46. Requisitos Funcionales relativos al Caso de Uso CU-26: Navegar entre las sentencias de un documento. . . . .	75
5.47. Requisitos Funcionales relativos al Caso de Uso CU-27: Importar modelo de aprendizaje preentrenado. . . . .	75
5.48. Requisitos Funcionales relativos al Caso de Uso CU-28: Exportar modelo de aprendizaje entrenado. . . . .	75
5.49. Requisitos Funcionales relativos al Caso de Uso CU-29: Activar asistencia con patrones de coincidencia. . . . .	76
5.50. Requisitos Funcionales relativos al Caso de Uso CU-30: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia. . . . .	76
5.51. Requisitos Funcionales relativos al Caso de Uso CU-31: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia. . . . .	77
5.52. Requisitos Funcionales relativos al Caso de Uso CU-32: Visualizar estadísticas de anotación del proyecto. . . . .	77
5.53. Requisitos Funcionales relativos al Caso de Uso CU-33: Visualizar resultados del entrenamiento de un modelo de aprendizaje. . . . .	78
5.54. Requisitos Funcionales relativos al Caso de Uso CU-34: Establecer opciones de importación. . . . .	78
5.55. Requisitos No Funcionales . . . . .	79
5.56. Reglas de Negocio . . . . .	80
6.1. Requisito de Información RI-E01: PROYECTO . . . . .	98
6.2. Requisito de Información RI-E02: CATEGORÍA_ENTIDAD . . . . .	98
6.3. Requisito de Información RI-E03: DOCUMENTO . . . . .	99

6.4.	Requisito de Información RI-E04: MODELO . . . . .	99
6.5.	Requisito de Información RI-E05: REGEX . . . . .	100
6.6.	Requisito de Información RI-E06: ANOTACIÓN . . . . .	100
6.7.	Requisito de Información RI-R01: DISPONER . . . . .	101
6.8.	Requisito de Información RI-R02: TENER . . . . .	101
6.9.	Requisito de Información RI-R03: ASOCIAR . . . . .	101
6.10.	Requisito de Información RI-R04: DESCRIBIR . . . . .	102
6.11.	Requisito de Información RI-R05: CONTENER . . . . .	102
6.12.	Requisito de Información RI-R06: PERTENECER . . . . .	102
6.13.	Diseño de Interfaz DI-01: Página de Inicio . . . . .	103
6.14.	Diseño de Interfaz DI-02: Página Principal de la Herramienta . . . . .	104
6.15.	Diseño de Interfaz DI-03: Página de Gestión de Expresiones Regulares . . . . .	105
7.1.	Etiquetas de Categorías Gramaticales - Part-of-speech tags . . . . .	124
7.2.	Tipos de Entidades Nombradas - Named Entity types . . . . .	125
8.1.	Prueba de Caja Negra PCN-01: Renombrar un proyecto de anotación existente. . . . .	132
8.2.	Prueba de Caja Negra PCN-02: Crear una nueva expresión regular. . . . .	132
8.3.	Prueba de Caja Negra PCN-03: Añadir una nueva anotación en un texto. . . . .	133
8.4.	Prueba de Caja Negra PCN-04: Editar los límites de una anotación existente. . . . .	133
8.5.	Prueba de Caja Negra PCN-05: Editar el color de una categoría de entidad. . . . .	134
8.6.	Prueba de Caja Negra PCN-06: Activar asistencia con patrones de coincidencia. . . . .	134
8.7.	Prueba de Caja Negra PCN-07: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia. . . . .	135
8.8.	Prueba de Caja Negra PCN-08: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia. . . . .	135
8.9.	Prueba de Caja Negra PCN-09: Importar fichero PDF. . . . .	136
8.10.	Prueba de Caja Negra PCN-10: Crear un proyecto de anotación. . . . .	136
8.11.	Prueba de Caja Negra PCN-11: Añadir una nueva categoría de entidad. . . . .	137
9.1.	Versiones de Paquetes Instalados . . . . .	143





# Resumen

Hoy en día, el formato textual es uno de los más frecuentes en el ámbito de la información. Un correcto tratamiento y anotación de los textos es fundamental para poder obtener los conceptos más relevantes asociados a ellos y así destacar lo que es realmente importante.

Los nuevos retos tecnológicos a los que nos afrentamos actualmente, tales como la transformación digital, requieren grandes cantidades de datos organizados y etiquetados correctamente para poder entrenar y hacer un uso eficiente de los modelos que subyacen a las nuevas implementaciones que se desarrollan.

Este proyecto se basa en la construcción de una herramienta de anotación de entidades nombradas con Procesamiento del Lenguaje Natural (PLN) para la asistencia en el etiquetado y categorización de las entidades descritas en documentos de texto digitalizados mediante Reconocimiento Óptico de Caracteres (OCR), permitiendo así agilizar la creación de extensos datasets de textos con sus entidades nombradas anotadas.

**Palabras claves:** Herramienta de Anotación, Procesamiento del Lenguaje Natural, Reconocimiento de Entidades Nombradas, Reconocimiento Óptico de Caracteres.





# Abstract

Nowadays, the textual format is one of the most frequent in the field of information. A correct treatment and annotation of the texts is essential to be able to obtain the most relevant concepts associated with them and thus highlight what is really important.

The new technological challenges we are currently facing, such as digital transformation, require large amounts of correctly organized and labeled data in order to train and make an efficient use of the models underlying the new implementations being developed.

This project is based on the construction of a named entity annotation tool with Natural Language Processing (NLP) to assist in the labeling and categorization of the entities described in digitized text documents using Optical Character Recognition (OCR), thus allowing to speed up the creation of large text datasets with their annotated named entities.

**Palabras claves:** Annotation Tool, Natural Language Processing, Named Entity Recognition, Optical Character Recognition.

# Capítulo 1

## Introducción

En la actualidad los datos son el recurso corporativo de mayor importancia. Éstos son utilizados por las organizaciones como medio para generar valor que reporte en beneficios para sus intereses.

El tratamiento de los datos se vuelve un aspecto muy importante que debe tenerse en cuenta, pues nos permitirá transformar las grandes cantidades de datos en información relevante y útil para el caso de uso que se pretenda desarrollar. Los datos que se generan son de tipos muy diversos, como los textuales, numéricos, binarios, etc. En concreto, en este proyecto nos centraremos en los datos de tipo textual.

Los textos son una rica fuente de información de dónde podemos extraer conocimiento útil para muchos aspectos de negocio. El etiquetado de partes concretas dentro de los textos permite destacar su información más relevante, así como conseguir estructurarlas para un tratamiento consistente. Estos son los fundamentos de la anotación, poder asignar una clase concreta a cada parte de un texto con el objetivo de clasificarla en una determinada categoría.

Existen diversos tipos de anotación aplicables sobre los textos, como son el etiquetado de partes del discurso (POS tagging), la anotación semántica o la anotación de entidades nombradas, cada uno de los cuales se aplicará a un propósito diferente. En este proyecto nos centraremos en este último tipo de anotación en el que se ve involucrada la tecnología de Reconocimiento de Entidad Nombradas (NER) [5].

La tecnología NER es una tarea de Extracción de Información (IE) perteneciente al campo de Procesamiento del Lenguaje Natural (PLN) [4] que consiste en la localización y etiquetado de entidades nombradas dentro de los textos para su clasificación en categorías de entidad definidas. Entre estas categorías de entidad se incluyen, comúnmente, los nombres de personas, organizaciones, lugares, emails, URLs, etc. De igual manera, pueden considerarse otras categorías de entidad distintas de acuerdo con las necesidades planteadas por el caso de uso con el que se trabaje. El funcionamiento de esta tecnología, en aras de su eficacia, plantea el uso de soluciones basadas tanto en patrones de coincidencia como en modelos de aprendizaje automático.

El creciente interés en torno a estas tareas de IE se ve motivada en parte por la evolución de la web y el proceso de transformación digital, los cuales han supuesto un claro avance tecnológico, con la información de tipo textual como eje principal. El auge de la Web 3.0 y la digitalización de documentos ofrecen nuevas oportunidades de negocio que requieren emplear tecnologías como

NER para sintetizar la nueva información y poder categorizar sus contenidos convenientemente.

La información textual que se recibe como entrada de un proceso de NER puede aparecer en distintos contextos y provenir de diferentes fuentes, lo cual obliga a hacer un tratamiento específico según el dominio concreto en el que se encuentre la información. En concreto, para la extracción de entidades a partir de imágenes con contenido textual o de ficheros PDF, primero es necesario realizar una etapa de preprocesamiento para adecuar la entrada al proceso de NER. La tecnología empleada para la derivación del texto contenido en estos elementos implica el uso de un motor de Reconocimiento Óptico de Caracteres (OCR) [6], cuya eficacia en el resultado obtenido repercutirá en cierta forma en la salida del proceso de NER. Esto debe tenerse en cuenta a la hora de aplicar NER en la extracción de las entidades nombradas dentro de documentos digitalizados. Si la transcripción llevada a cabo por el motor OCR no es buena, la tarea de NER tendrá una menor efectividad.

En cuanto al contexto al que pertenezcan los textos, no utilizaremos ni trataremos de igual manera textos provenientes de publicaciones en una red social que documentos escaneados de una entidad bancaria. Así mismo, el proceso para maximizar su valor y los fines para los que se utilicen serán también diferentes. Principalmente, las categorías de entidad que se utilicen para nombrar las entidades de un texto en un contexto concreto no servirán para textos de otro contexto distinto. Así mismo, los posibles modelos de NER que se entrenen para automatizar la extracción de sus entidades no serán igual de eficaces en la predicción de entidades en los textos de otros contextos distintos.

En definitiva, este proyecto se centra en la tarea de NER aplicada sobre documentos de texto, ya sean derivados a partir de un proceso previo de digitalización mediante OCR o simplemente vengan ya representados como texto plano. Las categorías de entidad con las que clasificar las entidades nombradas a anotar pueden ser definidas a conveniencia según el caso de uso concreto que se requiera, el cual describirá la clase de información relevante que se desea reconocer en los documentos, permitiendo obtener tanto datasets de documentos anotados personalizados como modelos de aprendizaje entrenados para predecir entidades nombradas en el contexto de documentos que se precise.

### 1.1. Motivación

La motivación principal para llevar a cabo la construcción de una herramienta de anotación es agilizar la creación de conjuntos de documentos (datasets) etiquetados con los que entrenar modelos de NER para la predicción de entidades nombradas en nuevos textos no anotados, los cuales sirvan a su vez como mecanismo para retroalimentar el propio proceso de anotación de los nuevos documentos. Para entrenar forma efectiva un modelo de aprendizaje es necesario disponer de grandes datasets de datos de ejemplo con los que entrenar y evaluar el modelo, para, en consecuencia, poder mejorarlo y acabar consiguiendo un modelo de utilidad. Estos datasets, además de tener numerosos ejemplos, deben ser adecuados para el caso de uso al que están destinados. Por ejemplo, si queremos un modelo para reconocer entidades en documentos de texto científicos será conveniente entrenarlo también con ejemplos del ámbito científico, y no con textos de poesías o de redes sociales. Además, lo más probable es que la clase de información

que queramos derivar de cada uno de ellos sea diferente.

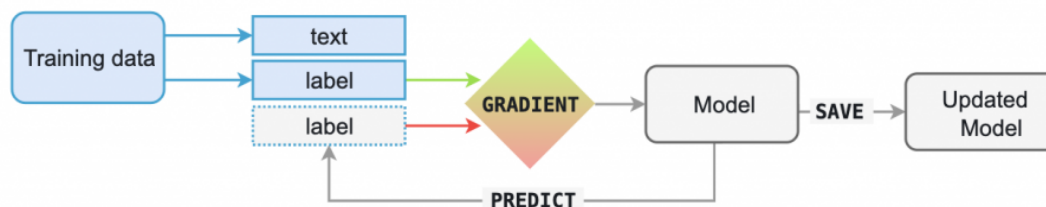


Figura 1.1: Proceso de entrenamiento de un modelo de aprendizaje

El proceso de entrenamiento mencionado que se persigue aparece resumido en la figura 1.1. Este proceso de entrenamiento para obtener un modelo de NER a partir de extensos datasets con datos de entrenamiento atiende a los fundamentos del Deep Learning [7], basado en el uso de redes neuronales con múltiples capas de neuronas de aprendizaje, las cuales tienen asociados unos pesos configurables que se van ajustando con el paso de los ejemplos de entrenamiento y que determinan las predicciones devueltas por el modelo. A medida que la red neuronal se vaya entrenando con los ejemplos de textos etiquetados, se irá reduciendo el valor de pérdida asociado a las predicciones del modelo para esos ejemplos de entrenamiento, debiendo continuar hasta lograr un modelo que clasifique correctamente la gran mayoría de los ejemplos de entrenamiento y, a su vez, generalice bien para nuevos ejemplos de textos con los que el modelo no ha sido entrenado. Por ello, es preciso fomentar el entrenamiento con grandes datasets de ejemplos diferentes que contribuyan a entrenar modelos de NER lo más efectivos posibles, sin perder su capacidad de generalización.

Debido a que el proceso de anotación requerido para conseguir extensos datasets de datos etiquetados es lento y tedioso si se hace de forma manual, se considera de utilidad añadir una capa de aprendizaje activo que retroalimente el proceso y haga predicciones de posibles anotaciones en el texto que el usuario pueda aceptar o rechazar en función de su corrección. Esta característica permitiría agilizar el proceso de anotación al no tener al usuario que anotar explícitamente gran parte de las entidades nombradas, sino solo verificarlas o corregirlas. A medida que avanza el proceso, el propio modelo iría aprendiendo de las anotaciones verificadas y haciendo más predicciones de entidades correctas. Por tanto, la retroalimentación proporcionada también por el usuario al corregir las predicciones de entidades del modelo es esencial para conseguir un modelo más eficiente. Así, se conseguiría tener en un menor tiempo un modelo de NER adecuado para extraer eficazmente las entidades nombradas de nuevos textos del mismo contexto donde fue entrenado y listo para ser usado en producción.

## 1.2. Contexto

Esta sección describe el contexto en el cual se desarrolla el proyecto, incluyendo los conceptos fundamentales relacionados con las partes del mismo. Se pretende dar unas nociones básicas

que ayuden a entender las diferentes tecnologías, procesos y metodología utilizada, de modo que se comprendan las bases teóricas y procedimentales que sustentan el proyecto.

Una especificación completa de cada uno de estos aspectos puede ser consultada en sucesivas secciones de la presente memoria.

### **Metodología de Trabajo**

La metodología escogida para estructurar y organizar el trabajo a realizar durante el proyecto ha sido **UVagile** [1], una propuesta de la comunidad universitaria de la Universidad de Valladolid basada en el marco de trabajo ágil Scrum.

UVagile es una metodología que pretende adaptar el uso de Scrum al ámbito académico, asimilando sus conceptos para aplicarlos en los proyectos de asignaturas y Trabajos de Fin de Grado. En lo que se refiere a este TFG, el trabajo ha sido planificado para ser realizado durante 5 Sprints de una duración de 3 semanas cada uno. Durante dichos períodos de tiempo se llevarán a cabo el conjunto de tareas necesarias para alcanzar los objetivos propuestos en el proyecto.

Para el seguimiento del proyecto se habilitan reuniones semanales (Weekly) en las que se comunica el trabajo realizado desde la anterior reunión, el trabajo a realizarse hasta la siguiente, así como los impedimentos que hayan podido surgir durante la semana. Por otra parte, siguiendo los preceptos del desarrollo ágil incremental, se finaliza cada Sprint con una reunión, llamada Retroplanning, en la que se lleva a cabo la entrega de un incremento de producto funcional, la retrospectiva para inspeccionar el trabajo del equipo y derivar posibles acciones de mejora y la planificación del siguiente Sprint del proyecto.

Estas reuniones llevan asociados los siguientes timebox de tiempo máximo para su realización:

- **Weekly:** máximo de 15 minutos.
- **Retroplanning:** máximo de 1 hora.

La adaptación de la metodología a este proyecto concreto se especifica en el Capítulo 3 de Planificación y Presupuesto del Proyecto.

Para clarificar el entendimiento del resto de partes involucradas en el proyecto se considera útil introducir unas breves descripciones sobre cada una.

### **Anotación**

La anotación es el proceso por el cual se etiqueta dentro de categorías la información relevante contenida en diferentes elementos, ya sean texto, imágenes, etc [3].

En este proyecto la anotación se circunscribe a la anotación de entidades nombradas en textos. El etiquetado de estas entidades se hace en torno a un conjunto de categorías definidas que pueden ser personalizadas y que variarán según el dominio de la información que se maneje. En nuestro caso, diferenciaremos entre anotación manual y anotación asistida, dependiendo de si

la anotación es realizada íntegramente por parte del usuario o si ésta se ve automatizada por un modelo de Reconocimiento de Entidades Nombradas.

## Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (Natural Language Processing) es un campo dentro de la Inteligencia Artificial y la Lingüística que trata el estudio del lenguaje natural para la construcción de modelos de lenguaje entendibles por los ordenadores. Su objetivo es el de transformar la información escrita o hablada por las personas para comunicarse en una representación de dicha información adecuada para ser procesada por un ordenador.

Dentro del Procesamiento del Lenguaje Natural destacan multitud de tareas específicas relacionadas con el lenguaje. En nuestro caso, nos centraremos en una de ellas, el Reconocimiento de Entidades Nombradas.

## Reconocimiento de Entidades Nombradas

El Reconocimiento de Entidades Nombradas (Named Entity Recognition) es una tarea dentro del área de Extracción de Información (Information Extraction) que consiste en la identificación y clasificación en categorías predefinidas de las entidades nombradas existentes en documentos de texto.

Para la implementación de esta tarea se construyen modelos de NER con los poder hacer predicciones de nuevas entidades nombradas en textos del dominio que se plantee. Para ello, existen 2 grandes aproximaciones, como son los **patrones de coincidencia** y el **entrenamiento de modelos de aprendizaje**. La combinación de ambas aproximaciones permitirá obtener resultados más precisos y mejorar la eficacia del modelo resultante.

Esta tecnología de Reconocimiento de Entidades Nombradas es la que nos permitirá aportar una capa de asistencia a la anotación de entidades en nuestra herramienta.

## Reconocimiento Óptico de Caracteres

El Reconocimiento Óptico de Caracteres (OCR) es una tecnología empleada para la digitalización de documentos de texto. Su finalidad es extraer el texto contenido en elementos como imágenes a través de un proceso de detección y reconocimiento de los caracteres contenidos en ellas llevado a cabo por el motor OCR.

Dentro de este proyecto, esta tecnología nos será de gran utilidad para extraer el texto contenido en ficheros PDF que el usuario decida importar a la herramienta. Aunque existen otras formas de extraer la información de ficheros PDF, esta aproximación es bastante eficiente al estar el motor OCR entrenado para reconocer múltiples tipos de fuentes de texto diferentes.

## 1.3. Visión y Alcance

### Visión del Producto

Annotator es una herramienta de anotación de entidades nombradas en textos que permite personalizar los tipos de entidades utilizados y agilizar la creación de conjuntos de ejemplos por medio de la asistencia por aprendizaje utilizando un modelo de NER, con el objetivo de solventar el problema de la escasez de datos etiquetados en el entrenamiento y evaluación de modelos de aprendizaje permitiendo construir rápidamente extensos datasets con datos etiquetados adecuados al caso de uso que se desee abordar.

### Alcance del Proyecto

La herramienta de anotación a construir se empleará para anotar entidades en documentos de textos, por lo que la anotación en otros formatos distintos al textual no estará soportada.

Para llevar a cabo el proceso de anotación se utilizará una aplicación web con interfaz de usuario. Esta interfaz de usuario permitirá el uso de ratón para agilizar el etiquetado de las partes del texto relevantes. También, dispondrá de opciones directas para facilitar la revisión, modificación y eliminación de las anotaciones realizadas y para activar diferentes modos de asistencia a la anotación. Dentro de estos modos de asistencia se encuentran la coincidencia con patrones de expresiones regulares, el entrenamiento de modelos a partir de anotaciones existentes o la predicción a partir de un modelo de NER importado.

En cuanto al dominio de anotación, la herramienta se limitará a la anotación de entidades nombradas en el texto, no considerando la anotación de posibles relaciones existentes entre las entidades de un mismo texto. Dicha característica ya pertenecería al dominio de otra tarea de PLN distinta.

Con el objetivo de ilustrar las características detalladas que se plantean como solución de nuestra herramienta Annotator se emplea el árbol de características que se muestra en la Figura 1.2.

Un árbol de características es un diagrama utilizado para especificar las características principales de las que se compone un producto. Este tipo de diagrama refleja dichas características a distintos niveles de anidación según el grado de abstracción y detalle empleado.

- El primer nivel especifica los aspectos de alto nivel del producto.
- El segundo nivel parte del anterior añadiendo características más detalladas.
- El tercer nivel se asocia ya con las historias de usuario concretas a implementar para el desarrollo del producto.



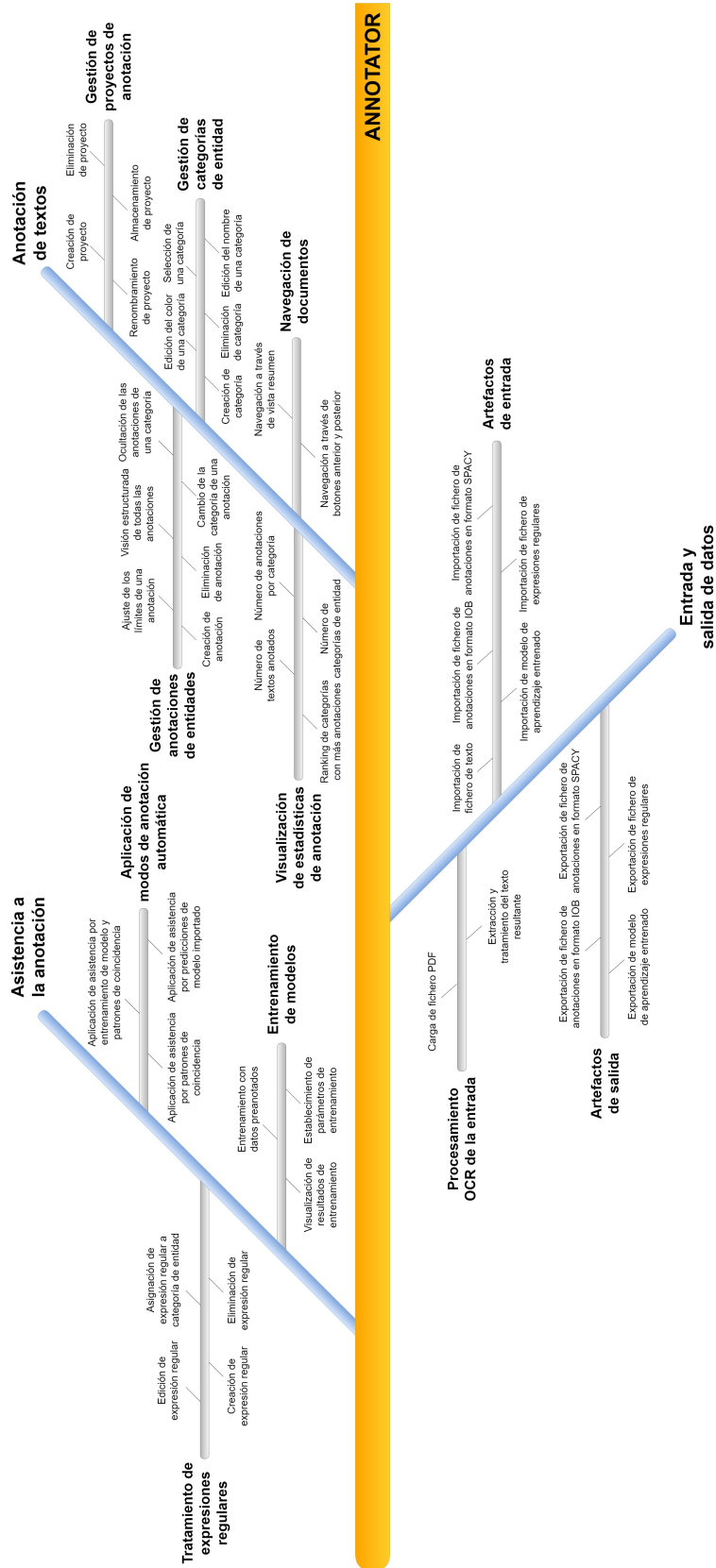


Figura 1.2: Árbol de Características de Annotator

## 1.4. Objetivos del Proyecto

A continuación, se formulan los objetivos principales que se pretenden conseguir con la realización de este proyecto.

ID Objetivo	Nombre Objetivo	Descripción Objetivo
OBJ-01	Digitalización de documentos de texto	Consiste en el proceso completo de digitalización de los documentos de texto, sobre los que se llevará a cabo el proceso de anotación, mediante la tecnología de Reconocimiento Óptico de Caracteres (OCR). Este es un objetivo independiente para proveer documentos en formato texto como entrada al proceso de anotación.
OBJ-02	Construcción de la herramienta de anotación	Consiste en la implementación de la lógica subyacente a la herramienta de anotación junto con su interfaz de usuario.
OBJ-03	Asistencia a la anotación	Consiste en la creación de modelos de aprendizaje automático personalizados con los que poder hacer predicciones para la anotación de entidades en textos de diferentes dominios.

Tabla 1.1: Objetivos del Proyecto

De los objetivos anteriores, el objetivo OBJ-02 incluye un subobjetivo relacionado con el estudio de las herramientas de anotación existentes, a partir del cual se derivan las características básicas que será interesante incorporar en nuestra herramienta.

ID Subobjetivo	ID Objetivo principal	OBJ-02
	Nombre Subobjetivo	Descripción Subobjetivo
OBJ-02.01	Estudio de las herramientas existentes	Consiste en el análisis de las herramientas de anotación de texto actuales. Incluye la realización de un estudio del funcionamiento y características ofrecidas por las herramientas existentes.

Tabla 1.2: Subobjetivo de OBJ-02

Cada uno de estos objetivos está asociado a unos determinados criterios de aceptación cuyo cumplimiento implicará su adecuada consecución. A continuación, dichos criterios de aceptación aparecen organizados según el objetivo de proyecto al que se adjuntan/relativo a ellos.

ID Criterio Aceptación	ID Objetivo asociado	OBJ-01
	Descripción Criterio de Aceptación	
CA-01	La herramienta permite importar un nuevo documento para digitalizar.	
CA-02	La herramienta implementa el proceso completo de digitalización de documentos.	
CA-03	La herramienta deriva un texto como salida del proceso de Reconocimiento Óptico de Caracteres (OCR).	

Tabla 1.3: Criterios de aceptación para OBJ-01

ID Criterio Aceptación	ID Objetivo asociado	OBJ-02
	Descripción Criterio de Aceptación	
CA-01	La herramienta permite visualizar las anotaciones realizadas.	
CA-02	La herramienta permite seleccionar entidades en el texto para su anotación.	
CA-03	La herramienta dispone de mecanismos para editar/eliminar una anotación.	
CA-04	La herramienta permite añadir una nueva categoría con la que anotar las entidades de un texto.	
CA-05	La herramienta permite volver a visitar un texto anotado anteriormente.	
CA-06	La herramienta permite guardar nuevas anotaciones para almacenarlas en el formato adecuado.	
CA-07	La herramienta permite visualizar estadísticas de anotación asociadas a un proyecto.	

Tabla 1.4: Criterios de aceptación para OBJ-02

## 1.5. Estructura del documento

Este apartado describe la organización del presente documento y resume el contenido de cada una de sus partes principales. Este documento se encuentra estructurado de la siguiente forma:

1. **Introducción:** este capítulo comienza describiendo brevemente la motivación que da sentido a la realización de este proyecto, así como el contexto en el cual se lleva a cabo. A continuación, establece la visión y alcance del producto que se va a construir, y termina citando los objetivos principales que se pretenden conseguir con la realización del proyecto.

ID Criterio Aceptación	ID Objetivo asociado	OBJ-02.01
	Descripción Criterio de Aceptación	
CA-01	El estudio detalla el funcionamiento de las herramientas analizadas.	
CA-02	El estudio describe las características asociadas a cada herramienta de anotación estudiada.	
CA-03	El estudio incluye una comparativa entre las herramientas de anotación estudiadas.	
CA-04	El estudio revisa las características de las herramientas existentes que pueden ser incorporadas a la herramienta de anotación a construir.	
CA-05	El estudio incluye unas conclusiones generales a partir de las herramientas estudiadas.	

Tabla 1.5: Criterios de aceptación para OBJ-02.01

ID Criterio Aceptación	ID Objetivo asociado	OBJ-03
	Descripción Criterio de Aceptación	
CA-01	La herramienta realiza predicciones de anotación de entidades a partir de un modelo de aprendizaje automático.	
CA-02	La herramienta permite la opción de entrenar un modelo de aprendizaje automático a partir de anotaciones validadas.	
CA-03	La herramienta dispone de mecanismos para validar o rechazar la anotación de un determinado texto.	
CA-04	La herramienta permite activar distintos modos de asistencia a la anotación.	
CA-05	La herramienta permite visualizar resultados del entrenamiento de un modelo de aprendizaje automático.	

Tabla 1.6: Criterios de aceptación para OBJ-03

2. **Estado del arte:** este capítulo analiza las herramientas de anotación existentes en la actualidad, abarcando tanto sus características más destacadas como su propio funcionamiento práctico.
3. **Planificación y Presupuesto:** este capítulo establece la metodología con la que se ejecuta el proyecto, expone la planificación temporal planteada y explica los cálculos realizados a la hora de llevar la estimación del proyecto y determinar el presupuesto total del mismo.
4. **Fundamentos Teóricos:** en este capítulo se profundiza en las bases teóricas asociadas a

las principales tecnologías y componentes del dominio del proyecto.

5. **Análisis:** este capítulo identifica el dominio en el que se enmarca el proyecto y describe los diferentes requisitos asociados a nivel de Historias de Usuario. Además, se incluyen dentro de este análisis los principales casos de uso identificados junto con requerimientos de otros tipos, como son los funcionales, no funcionales o las reglas de negocio.
6. **Diseño:** este capítulo describe de forma pormenorizada los diferentes componentes de la etapa de diseño del producto, destacando aspectos relativos al modelo de datos utilizado en la aplicación y al diseño de la interfaz de usuario, así como a las arquitecturas lógica y física correspondientes.
7. **Implementación:** este capítulo se circunscribe a la parte de codificación del proyecto en la que se implementa la funcionalidad requerida para el producto e incluye una descripción detallada de las tecnologías involucradas y herramientas utilizadas durante su desarrollo.
8. **Manuales:** en este capítulo se incluyen los manuales de usuario y de instalación de la herramienta.
9. **Conclusiones y trabajo futuro:** este último capítulo incluye una conclusión tanto a nivel de proyecto como de producto y, además, expresa algunas ideas sobre las posibles vías que se podrían tomar para una futura evolución del producto.



# Capítulo 2

## Estado del Arte

En estos últimos tiempos, la evolución de las técnicas de aprendizaje automático ha derivado en lo que conocemos como Deep Learning [7], siendo aplicable cada vez a más casos de uso. La mayoría de técnicas y algoritmos de Deep Learning basan su efectividad en el procesamiento de grandes conjuntos de datos etiquetados, lo cual ha llevado a la necesidad de disponer de un mecanismo eficiente para la anotación de los datos con los que entrenar los modelos de aprendizaje.

Es por ello que las herramientas de anotación han proliferado y se muestran como una solución de gran utilidad para etiquetar datos de distinto tipo de una forma rápida y sencilla. A pesar de ello, no todas las herramientas de anotación actuales hacen uso de estas técnicas de aprendizaje para aplicarlas en el proceso de anotación.

Dentro de las herramientas de anotación utilizadas actualmente veremos las siguientes, cuyo funcionamiento y características se detallan a continuación.

### 2.1. Prodigy

Esta es una herramienta de anotación de propósito general que abarca desde el reconocimiento de objetos en imágenes y la clasificación global del contenido de un documento hasta el reconocimiento y anotación de las entidades nombradas dentro de textos junto con sus posibles relaciones.

Desde el punto de vista de este proyecto, nos centraremos solo en la parte destinada a la anotación de entidades en documentos de texto.

Prodigy es una herramienta moderna de anotación que tiene la característica de asistencia a la anotación por medio de un modelo de aprendizaje que simplifica y agiliza el proceso general de anotación. La empresa que desarrolló esta herramienta es Explosion, muy reconocida en el sector tecnológico por otros productos como las librerías de aprendizaje Spacy (orientada al Procesamiento del Lenguaje Natural) y Thing (orientada a Deep Learning), lo cual da cierta credibilidad al resto de sus productos.

Prodigy permite hacer tanto anotación manual sin asistencia a la anotación, como anotación asistida, utilizando el aprendizaje activo mencionado anteriormente. En esta segunda modalidad, permite al usuario anotador corregir posibles fallos de predicción del modelo de reconocimiento

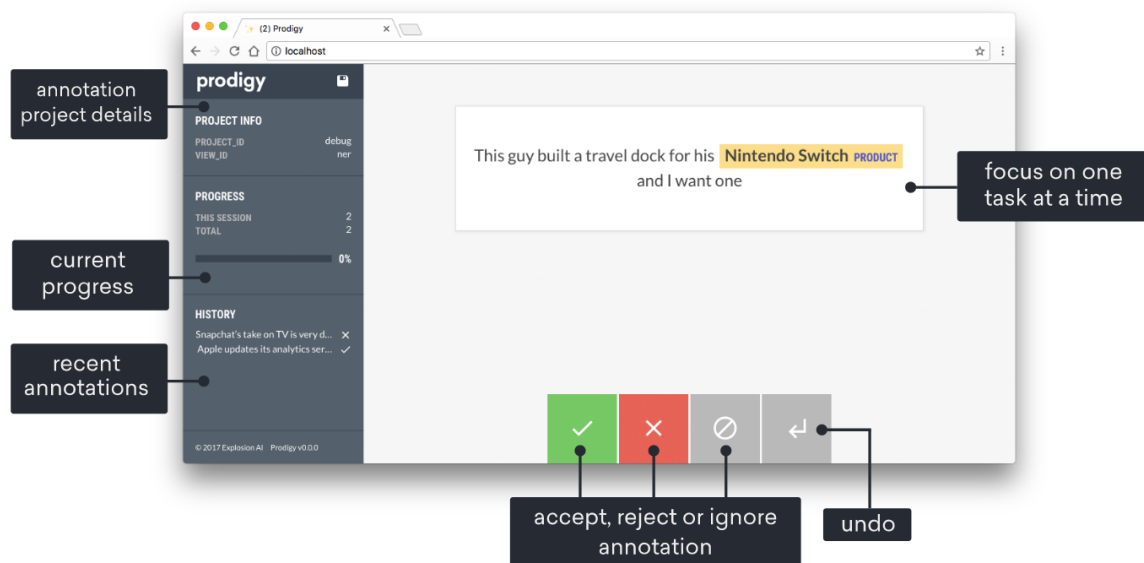


Figura 2.1: Interfaz Herramienta Prodigy

de entidades o, simplemente, etiquetar nuevas entidades que el modelo no ha considerado. Esas correcciones y nuevas anotaciones retroalimentan el modelo de PLN utilizado, permitiendo evaluar progresivamente su rendimiento y, en consecuencia, pudiéndolo mejorar.

Uno de los casos de uso ampliamente extendidos es crear un dataset con las anotaciones realizadas con Prodigy e incorporarlo a Spacy para poder entrenar y evaluar un modelo de PLN rápidamente.

Concretamente, para la tarea de Reconocimiento de Entidades Nombradas, Prodigy dispone de comandos específicos que incorporan la característica de aprendizaje activo a través de modelos de PLN preentrenados propios de Spacy, la librería para PLN. Así mismo, permite utilizar modelos de PLN personalizados por el propio usuario con el objetivo de poder evaluarlos y mejorarlos. La integración directa de Prodigy con los modelos de Spacy es una de sus grandes ventajas para agilizar el proceso de anotación de una forma muy sencilla.

Toda esta funcionalidad descrita está contenida en 3 comandos principales:

- **ner.manual**: para la anotación manual de un texto.
- **ner.correct**: para la corrección manual de las predicciones realizadas por el modelo.
- **ner.teach**: para la actualización y mejora del modelo.

Por otra parte, Prodigy es una herramienta de anotación que se descarga en el propio ordenador previo pago de una licencia propietaria y que no interactúa con servicios de cloud para alojar las anotaciones realizadas o modelos entrenados. El hecho de que no aloje ningún dato fuera del equipo del usuario contribuye a la protección de la privacidad de las anotaciones al



no almacenarlas en ningún servidor de Internet. Una vez descargada, puede ser instalada en un entorno virtual de Python e importada para su uso en scripts de Python.

Es una herramienta concebida para el uso en equipos de trabajo ágiles con pocos miembros. Por ello, Prodigy, al contrario que otras herramientas, no provee funcionalidad para la gestión avanzada de proyectos de anotación entre equipos de trabajo o la administración de cuentas de usuario.

## Características

Sus características más destacadas se podrían resumir en las siguientes:

- Asistencia a la anotación a través de modelos preentrenados de Spacy.
- Posibilidad de corrección de las predicciones realizadas por el modelo de aprendizaje.
- Capacidad de anotación colaborativa en equipos de trabajo reducidos.
- Uso de ratón para gestionar la anotación de las entidades en la interfaz de usuario.
- Anotación de relaciones entre las entidades.
- Posibilidad de anotación en datos de diversos tipos, además de textos.

## 2.2. Doccano

Doccano es una herramienta open source de anotación exclusiva para textos que permite etiquetar partes de un texto a través de una interfaz web muy similar en apariencia a la de la anterior herramienta, (ver Sección 2.1). A diferencia de Prodigy, Doccano no tiene soporte para otro tipo de datos que no sean el textual.

Con esta herramienta se pueden construir datasets para distintas tareas de Procesamiento del Lenguaje Natural como Reconocimiento de Entidades Nombradas, Análisis de Sentimientos o Clasificación de Textos, entre otras.

Por otra parte, Doccano puede ser instalada en un entorno Python u ofrecida como un contenedor (container), lo que posibilita su despliegue directo en entornos cloud como Microsoft Azure, Amazon Web Services o Heroku.

En cambio, una de sus principales desventajas es que no dispone de asistencia al aprendizaje mediante modelos de aprendizaje, de modo que el usuario no recibirá sugerencias de anotación por parte de la herramienta y simplemente se limitará a realizar las anotaciones de forma manual. Debido a ello, el proceso de anotación con esta herramienta se ve ralentizado, sobre todo cuando necesitamos construir datasets con muchos ejemplos para entrenar nuestros modelos.

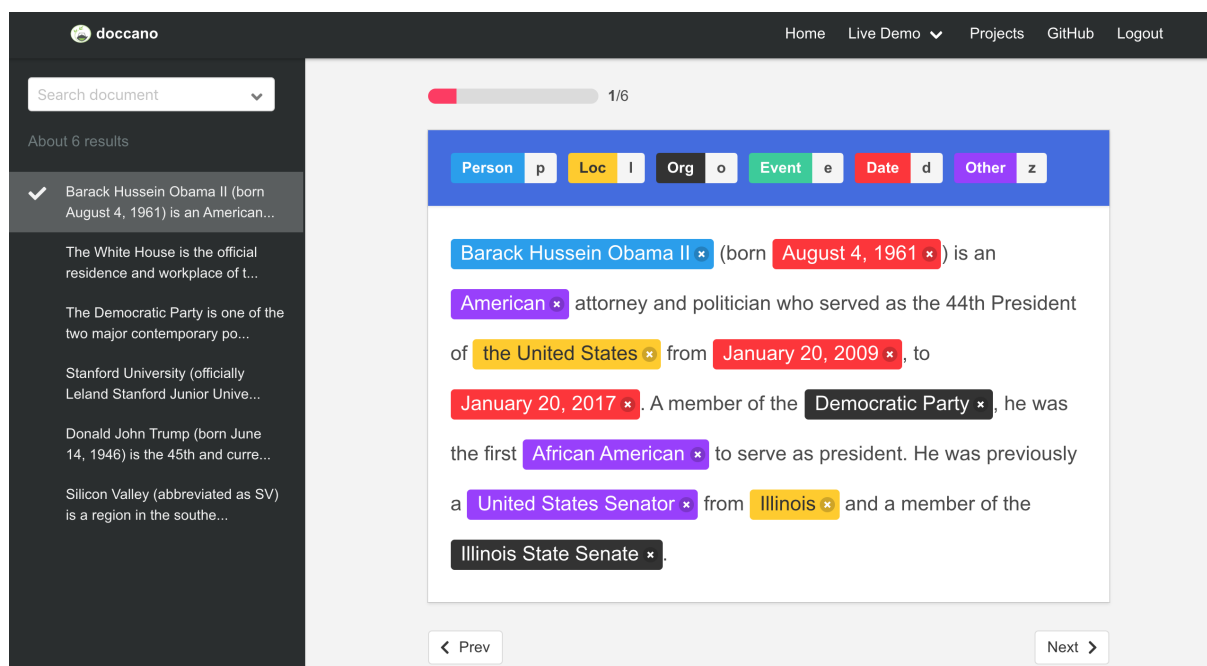


Figura 2.2: Interfaz Herramienta Doccano

## Características

Sus características más destacadas se podrían resumir en las siguientes:

- Capacidad de anotación colaborativa en equipos de trabajo reducidos.
- Uso de ratón para gestionar la anotación de las entidades en la interfaz de usuario.
- Adaptación de uso en dispositivo móvil.
- Existencia de una API REST propia a la que hacer peticiones.
- Posibilidad de despliegue en entornos cloud con su versión contenerizada.

## 2.3. Anafora

Anafora es una herramienta open source de anotación web aplicada a datos en formato texto pensada para estar centralizada en un servidor y ser usada por usuarios finales accediendo a la herramienta en dicho servidor. Su uso principal es la anotación de las entidades existentes en un texto, junto con sus posibles relaciones.

Anafora permite gestionar proyectos de anotación a gran escala en los que trabajen numerosos miembros de equipos de trabajo. Los anotadores accederían al servidor para crear las anotaciones, las cuales se almacenarían en el mismo servidor que actuaría a modo de repositorio central.

The screenshot shows the Anafora interface with a text document titled "allergycases\_org\_female-with-asthma-and-allergic" and user "stylerw". The document text is annotated with various entities and relations. A sidebar on the left shows a tree view of the schema, including TemporalEntities (EVENT, TIMEX3, DOCTIME, SECTIONTIME) and TemporalRelations (TLINK, ALINK). The main text area shows a paragraph about a 27-year-old female with asthma and her medical history. A detailed view of an annotation is shown on the right, including its ID, text, and a table of properties.

Relation Type	Relation
TLINK	[Source: in the fall], [Type: CONTAINS], [Target: nose]
TLINK	[Source: 6 months ago], [Type: CONTAINS], [Target: stopped]
TLINK	[Source: 2 times per week], [Type: OVERLAP], [Target: symptoms]
TLINK	[Source: in the past year], [Type: CONTAINS], [Target: visits]

Name	Value
DocTimeRel	OVERLAP
Type	N/A
Degree	N/A
Polarity	POS
ContextualModality	ACTUAL
ContextualAspect	N/A
Permanence	UNDETERMINED

Figura 2.3: Interfaz Herramienta Anafora

Esta arquitectura implica la necesidad de disponer de un servidor, concretamente uno basado en UNIX, y, además, una conexión estable a Internet para poder acceder al servidor central sin complicaciones.

Por otra parte, Anafora está pensada para ser simple y sencilla de usar por los usuarios finales. Así es que tanto los esquemas de las anotaciones como los propios datos de anotaciones generadas se almacenan en formato XML, un formato de representación de datos dirigido a ser fácilmente examinado y/o modificado por humanos. Al igual que las anteriores herramientas mencionadas, Anafora dispone de facilidades de interfaz de usuario para facilitar la anotación orientadas al uso del ratón y de atajos de teclado.

Además, Anafora dispone de la característica de adjudicación para supeditar las anotaciones realizadas por más de 1 usuario final sobre un mismo documento. Para ello, define el rol de adjudicador, el cual se encargará de verificar las anotaciones realizadas así como de resolver los posibles conflictos que pudieran surgir. Por lo tanto, el adjudicador toma un rol de moderador a la hora de confirmar la corrección de un documento anotado por otros usuarios.

## Características

Sus características más destacadas se podrían resumir en las siguientes:

- Capacidad de gestión de la anotación colaborativa en proyectos orientados a grandes equipos de trabajo.
- Uso de ratón para gestionar la anotación de las entidades en la interfaz de usuario.

- Anotación de relaciones entre las entidades.
- Característica y rol para la adjudicación de documentos anotados por varios usuarios finales.

## 2.4. Brat

Brat es una herramienta open source de anotación de documentos de texto destinada a tareas como la anotación de relaciones entre entidades, la extracción de eventos o la resolución de correferencias, además de para la anotación de entidades nombradas. Esta herramienta puede ser descargada a partir de su código fuente libre y ejecutada en el propio ordenador del usuario.

Brat puede integrar en el proceso de anotación la salida producida por otras herramientas de anotación automática externas, aunque no tiene la capacidad para entrenarlos a medida que se lleva a cabo el proceso de anotación. Por tanto, no se considera que tenga anotación asistida como tal.

Una de sus características es que permite estructurar la información disponible de una manera más consistente que las soluciones vistas anteriormente, a través del uso de colecciones para agrupar y organizar los documentos de textos sobre los que se hacen las anotaciones. También, permite la creación de cuentas de usuario para gestionar los permisos de edición y acceso a las colecciones y documentos. Esta característica puede resultar de utilidad en equipos de trabajo pequeños cuyos miembros se encargan de la anotación de distintos documentos.

Además, las anotaciones de entidades en Brat pueden tener asociada mayor cantidad de información para describir detalles sobre instancias concretas o enlazar a otra información relacionada. Por tanto, Brat está destinada a un proceso de anotación más exhaustivo y completo, pues no se limita simplemente a etiquetar las entidades de un texto sino que da posibilidad al usuario de incluir también ciertos detalles.

Por otra parte, al igual que Prodigy y Anafora, permite establecer anotaciones de tipo relaciones entre las entidades de un texto.

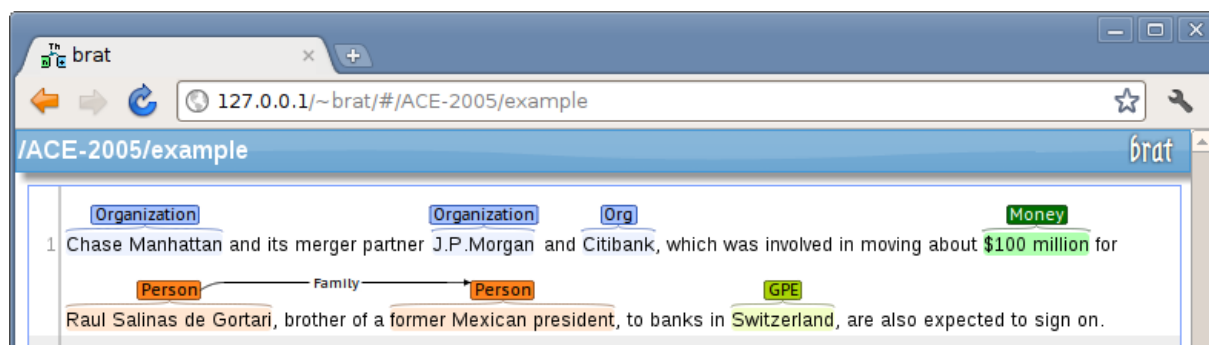


Figura 2.4: Interfaz Herramienta Brat

## Características

Sus características más destacadas se podrían resumir en las siguientes:

- Uso de colecciones para organizar los documentos de texto con las anotaciones.
- Uso de ratón para gestionar la anotación de las entidades en la interfaz de usuario.
- Anotación de relaciones entre las entidades.
- Gestión de cuentas de usuario para establecer permisos de edición de documentos.

## 2.5. Tagtog

Tagtog es una herramienta de anotación de textos con una parte de su funcionalidad de uso gratuito y otras características adicionales de pago, todo ello según la forma de uso. Respecto a ello, puede ser utilizada a través de Internet (por medio de la creación de una cuenta de usuario) o directamente en la infraestructura del cliente (On-Premises). Su versión gratuita solo está disponible para su uso en la nube a través de una cuenta de usuario, por lo que siempre se requerirá conexión a Internet para su uso gratuito.

Esta herramienta permite la anotación manual y asistida por aprendizaje automático de entidades nombradas, relaciones entre entidades o la categorización de documentos completos. Recientemente ha añadido también la funcionalidad de anotar directamente ficheros PDF.

The screenshot displays the Tagtog web interface. At the top, there is a navigation bar with 'Settings', 'Documents', 'Learning', and 'Downloads'. Below this is a toolbar with icons for document management and editing. The main document area shows a text snippet titled 'Automethylation of CARM1 allows coupling of transcription and mRNA splicing.' The text contains several entities highlighted in green (CARM1) and red (post-translational modifications, pre-mRNA splicing). To the right, a sidebar contains 'Document Labels' with dropdown menus for 'technical\_dept' and 'sales\_dept', and an 'Entity Tally' section showing counts for various entities.

Document Labels

show only defined

technical\_dept

?

sales\_dept

?

Entity Tally

unique 6 total 16

unidentified 5

CARM1	0/8
Coactivator-associated arginine methyltransferase 1	0/1
CARM1's	0/2
post-translational modifications	1/1

Figura 2.5: Interfaz Herramienta Tagtog

La versión gratuita Tagtog solo permite la anotación manual y su posible integración con modelos de aprendizaje del usuario para anotar automáticamente las entidades de un nuevo texto con el modelo. Esta integración se debe realizar por medio de un mecanismo de notificaciones a través de HTTP para conectar nuestro modelo con la herramienta Tagtog en la nube. Su característica de anotación automática con modelos de NER preentrenados por Tagtog solo está disponible con un plan de pago.

En cuanto a su interfaz de usuario, dispone de facilidades de ratón como ayuda al usuario para anotar las entidades que requiera.

Esta herramienta está pensada para ser utilizada por más de 1 persona, permitiendo gestionar sus anotaciones realizadas sobre los documentos. También, como medida de control de posibles inconsistencias que puedan producirse entre las anotaciones en un mismo texto, Tagtog dispone de la característica de adjudicación por la cual se pueden revisar las anotaciones realizadas por cada persona en un mismo documento y decidir cuáles son correctas. Posteriormente, las anotaciones realizadas pueden ser exportadas a un formato JSON, destacado por su flexibilidad y portabilidad.

Por otra parte, dispone de una API REST que permite hacer peticiones para importar a la herramienta textos sin anotar o exportar documentos ya con sus anotaciones.

### **Características**

Sus características más destacadas se podrían resumir en las siguientes:

- Integración con modelos de aprendizaje del usuario para la asistencia a la anotación.
- Capacidad de adjudicación en la anotación colaborativa por parte de varias personas.
- Uso de ratón para gestionar la anotación de las entidades en la interfaz de usuario.
- Anotación de relaciones entre las entidades.
- Disponibilidad de una API REST para gestionar los documentos y sus anotaciones.

## **2.6. Comparativa Herramientas de Anotación**

A continuación, se presenta una matriz resumen con las principales características elegidas para el objetivo de realizar una comparación entre las distintas herramientas de anotación estudiadas.

Así mismo, en esta comparativa se incluye también Annotator, nuestra propia herramienta de anotación, especificando las características deseables que se espera que implemente. No se espera que la herramienta Annotator implemente todas las características mostradas por las demás herramientas, pues su desarrollo debe ajustarse a la planificación de un proyecto de TFG.

<i>Herramienta Estudiada</i>	<b>Gratuidad</b>	<b>Asistencia a la Anotación</b>	<b>Gestión para Colaboración en Equipos de Trabajo Grandes</b>	<b>Anotación de Relaciones entre Entidades</b>	<b>Sin Requisitos de Conexión a Internet</b>	<b>Facilidades de Interfaz de Usuario</b>
<b>Annotator</b>	✓	✓	✗	✗	✓	✓
<b>Prodigy</b>	✗	✓	✗	✓	✓	✓
<b>Doccano</b>	✓	✗	✗	✗	✓	✓
<b>Anafora</b>	✓	✗	✓	✓	✗	✓
<b>Brat</b>	✓	✗	✗	✓	✓	✓
<b>Tagtog</b>	✓	✓	✓	✓	✗	✓

Tabla 2.1: Matriz Comparativa Herramientas de Anotación

## Conclusión

En conclusión, Prodigy es una de las herramientas de anotación más potentes entre las estudiadas, pues ofrece facilidades para agilizar el proceso de anotación (como la asistencia/active learning) y se integra de una forma directa con Spacy, una de las librerías más potentes para el procesamiento de lenguaje natural en Python.

Por otra parte, Tagtog también es una de las herramientas más completas variando las funcionalidades que ofrece según el plan que se contrate. Con respecto a las otras herramientas tiene los inconvenientes de no disponer de soporte para la gestión de proyectos en equipo de trabajo grandes y de ser una herramienta de pago, por lo que la inversión inicial a realizar será más interesante si se prevé desarrollar más proyectos relacionados con los casos de uso que cubre Prodigy.

Por otro lado, tanto Doccano, Anafora como Brat son herramientas de anotación gratuitas y de gran utilidad en el caso de tan solo necesitar anotar entidades en documentos de texto. En el caso de Anafora, su uso también puede ser bastante recomendado para equipos de trabajo grandes que quieran trabajar colaborativamente en las anotaciones de una manera centralizada. Igualmente, el funcionamiento de estas otras herramientas es sencillo e intuitivo y cualquier usuario con un conocimiento básico en informática podría utilizarlas.

Tras el análisis realizado sobre estas herramientas actuales, para la herramienta de anotación que implementaremos se considera relevante la posibilidad de incorporar aspectos como los siguientes:

- Asistencia a la anotación a través del uso de un modelo de aprendizaje: para proporcionar predicciones de anotación de entidades y así agilizar el proceso de anotación requerido.
- Creación de nuevos tipos de entidades nombradas: para poder personalizar los datasets de anotaciones al caso de uso concreto que se requiera.
- Exportación de anotaciones y modelos: para poder exportar los datasets de anotaciones creados y los modelos de NER entrenados.

- **Facilidades de Interfaz de Usuario:** para permitir una anotación más cómoda y efectiva al usuario. En esta categoría se incluyen el uso de ratón para gestionar las anotaciones o de colores para diferenciar los tipos de entidades, entre otros.
- **Vista de acceso a las anotaciones:** para poder visualizar, modificar o eliminar anotaciones realizadas anteriormente.
- **Vista resumen con resultados del proceso de anotación:** para poder sacar conclusiones valiosas acerca de las anotaciones obtenidas. Por ejemplo, rankings con los tipos de entidades más recurrentes en los documentos de texto anotados.



# Capítulo 3

## Planificación y Presupuesto

Esta sección comenzará explicando los fundamentos acerca de la metodología de desarrollo utilizada en el proyecto, así como la adaptación de dicha metodología al trabajo a realizar en este proyecto concreto. Seguidamente, se llevarán a cabo las estimaciones sobre las tareas planteadas para lograr los objetivos del proyecto y se planteará la planificación temporal del mismo. Finalmente, se terminará por determinar el presupuesto económico total del proyecto a partir de la valoración de los aspectos de hardware, software y recursos humanos necesarios.

### 3.1. Metodología Utilizada

La metodología escogida para estructurar y organizar el trabajo a realizar durante el proyecto ha sido **UVagile** [1]. Esta es una metodología creada dentro de la comunidad de la Universidad de Valladolid basada en Scrum, que pretende llevar los conceptos de este marco de trabajo ágil al ámbito académico.

Dentro del área de Trabajos de Fin de Grado, UVagile tiene como objetivo adaptar su modo de trabajo a la forma de trabajar en proyectos profesionales con equipos ágiles. Ello requiere un esfuerzo previo por aprender y entender los principios básicos que sustentan el marco de trabajo del que se deriva, Scrum.

Por lo tanto, se proceden a explicar los fundamentos principales de Scrum. Su funcionamiento se basa en la existencia de una serie de componentes, los cuales pasamos a detallar a continuación. Así mismo, una visión completa del marco Scrum se puede contemplar en la Figura 3.1.

#### Pilares

Scrum se enfoca en el empirismo del proceso de desarrollo del proyecto, definiendo 3 pilares básicos que son esenciales para su correcta implantación.

- **Transparencia:** basada en la visibilidad tanto del proceso que se lleva a cabo como de los artefactos que se generan y actualizan durante el mismo. Es clave tener una visión y entendimiento común entre los miembros del equipo en todos los aspectos.

- **Inspección:** basada en la revisión continua de los artefactos del proyecto y del propio proceso para poder mejorar y detectar posibles desviaciones indeseadas.
- **Adaptación:** basada en el ajuste del proceso para llevar el producto resultante a los ideales buscados. Una vez detectadas las desviaciones durante la inspección se acometen los cambios necesarios para hacer el ajuste.

Los eventos contenidos dentro de los sprints del proyecto pretenden la consecución de estos pilares. La coexistencia de estos 3 pilares durante el proyecto es fundamental para su éxito.

### Valores

Los valores son conceptos que los miembros de un proyecto deben adquirir para contribuir a cumplir con los pilares anteriores.

Estos valores mencionados son los siguientes:

- **Coraje:** por el que los miembros del equipo muestran aptitud para hacer su trabajo correctamente y enfrentarse a los problemas que surgan.
- **Focalización:** por la que los miembros del equipo se centran en el trabajo encomendado y los objetivos propuestos.
- **Compromiso:** por el que los miembros del equipo se responsabilizan de su trabajo para contribuir a los objetivos del equipo.
- **Respeto:** por el que cada miembro del equipo respeta al resto, tratando a todos por igual.
- **Apertura:** por la que los miembros del equipo se muestran abiertos a nuevos cambios u oportunidades que puedan surgir.

Generalmente, estos valores se pueden considerar soft skills que toda persona que trabaje en proyectos ágiles debe interiorizar.

### Roles

En Scrum conviven 3 roles distintos dentro de un proyecto. El concepto de equipo Scrum (Scrum Team) engloba a estos 3 roles de trabajo, teniendo cada uno unas responsabilidades y funciones específicas para llevar al equipo a conseguir los objetivos marcados.

Estos son roles son los siguientes:

- **Propietario del Producto (Product Owner):** es la persona encargada de maximizar el valor de lo que producen los desarrolladores al hacer su trabajo. Su principal responsabilidad es gestionar la Pila del Producto (Product Backlog) definiendo qué requisitos deben considerarse para el producto.

- **Desarrolladores (Developers):** es el conjunto de profesionales que se encargan de realizar el trabajo necesario para convertir una idea en un producto funcional y entregable al cliente. Este equipo de desarrolladores debe ser auto-organizado, multifuncional y diversificado para poder ejecutar su trabajo de forma efectiva y producir incrementos de producto con valor. Además, la responsabilidad final recaerá sobre el equipo de desarrolladores como un todo.
- **Scrum Master:** es la persona responsable de llevar a efecto el proceso de Scrum en una organización. Es el encargado de apoyar el uso de Scrum y ejerce de entrenador/facilitador del proceso para que este marco de trabajo sea asumido e interiorizado por todo el equipo Scrum. Su función no es ordenar al resto de miembros del equipo sino ser un líder al servicio del mismo. Esto hace que sean en él imprescindibles las soft skills, sabiendo tratar con las personas.

Lo ideal es el trabajo colaborativo entre todos los roles dentro del Equipo Scrum, procurando la comunicación y cohesión entre ellos para alcanzar los objetivos propuestos para el equipo.

Por otra parte, el tamaño sugerido para el equipo Scrum varía entre los 5 y 11 miembros, siendo más recomendable un número bajo.

#### Eventos

Los eventos en Scrum son períodos de tiempo limitado (time-boxes) con los que se busca afianzar una regularidad en el proceso Scrum evitando así la necesidad de programas reuniones no definidas. Dentro de ellos se llevan a cabo todas las tareas involucradas en un proyecto.

A continuación, se describen los eventos prescritos por Scrum para abarcar todo el trabajo de un proyecto.

- **Sprint:** es el evento principal en Scrum que actúa como contenedor del resto. Su función es la de englobar todo el trabajo necesario para construir un incremento de producto.  
Un proyecto se compone de varios sprints de igual duración en los que se producen siempre todos los eventos, favoreciendo la predictibilidad del proceso y creando una rutina de trabajo.
- **Sprint Planning (Planificación del Sprint):** este es evento en el que participa el equipo Scrum al completo y consiste en la planificación de todo el trabajo a realizar a lo largo del sprint. En cada Sprint Planning los desarrolladores definen el Sprint Backlog, el cual contiene las descripciones de las tareas que se completarán en el transcurso del Sprint. También, se define el objetivo del Sprint (Sprint Goal) que establece el alcance del incremento de producto a construir en el Sprint. Los desarrolladores utilizan el Sprint Goal como guía para determinar qué tareas llevar al Sprint Backlog.
- **Daily Scrum (Scrum Diario):** esta es una reunión diaria de corta duración (máximo de 15 minutos) en la que los desarrolladores examinan el progreso que cada uno de ellos ha experimentado durante el día anterior, planifican el trabajo a realizar durante el día en curso y comentan los impedimentos que han surgido.

Tiene como objetivo principal la comunicación entre los desarrolladores para crear dinámica entre ellos, así como para habilitar el pilar de inspección y evaluar el progreso del trabajo a completar durante el Sprint.

- **Sprint Review (Revisión del Sprint):** este evento tiene lugar al final de cada Sprint y consiste en la revisión del incremento de producto construido en el mismo con el Product Owner y algunos interesados (stakeholders). Se basa en la colaboración entre todos los asistentes para determinar qué nuevas funcionalidades se pueden añadir con vistas al siguiente incremento de producto.

Su principal objetivo es recibir retroalimentación (feedback) por parte de los interesados para poder optimizar el valor entregado en el producto.

- **Sprint Retrospective (Retrospectiva del Sprint):** es la última reunión del Sprint consistente en la evaluación del trabajo realizado por el equipo Scrum durante el Sprint.

Su objetivo es inspeccionar y adaptar la conducta del equipo durante el Sprint, para así mejorar en el siguiente. La mecánica de esta reunión se basa en cuestionar qué se hizo bien, qué se hizo mal y qué posibles mejoras se pueden implementar para el siguiente Sprint.

### Artefactos

Los artefactos en Scrum representan los objetos de valor en el proyecto sobre los que se trabaja y proporcionan la transparencia necesaria en el proceso para habilitar la inspección y adaptación del trabajo.

Los artefactos incluidos en la Guía Scrum [2] son los 3 siguientes, los cuales se pasan a detallar.

- **Product Backlog (Pila del Producto):** es una lista ordenada y priorizada dinámicamente que contiene todo lo necesario para ser incluido en el producto. Representa la única fuente de requisitos del proyecto y debe ser gestionada activamente por el Product Owner para actualizar la visión del producto. El nivel de detalle de cada uno de sus elementos varía, siendo mayor a medida que su implementación está más cercana en el tiempo.
- **Sprint Backlog (Pila del Sprint):** es el conjunto de elementos del Product Backlog seleccionados para ser implementados por los desarrolladores a lo largo del sprint. Constituye una predicción del trabajo que los desarrolladores consideran que son capaces de realizar durante el Sprint en curso. Los elementos que lo forman representan el trabajo necesario para cumplir con el objetivo propuesto para el Sprint. Es un artefacto gestionado íntegramente por los desarrolladores de forma activa que proporciona una visión del estado actual del Sprint.
- **Product Increment (Incremento de Producto):** es la suma de todos los elementos del Product Backlog implementados por los desarrolladores hasta el momento. Representa el

valor conseguido durante el Sprint, integrado junto con el resto de incrementos producidos en los Sprints anteriores.

Un incremento de producto debe alinearse con el concepto de Definition of Done (Definición de Hecho) que define lo que debe cumplir el incremento de producto para ser considerado como completado. Esta definición puede ser representada como una lista de criterios de aceptación que el incremento debe satisfacer para ser considerado como realizado. Es importante que todos los miembros del Equipo Scrum tengan un entendimiento compartido en lo que se refiere a Definición de Hecho para así cumplir con el pilar de la transparencia.

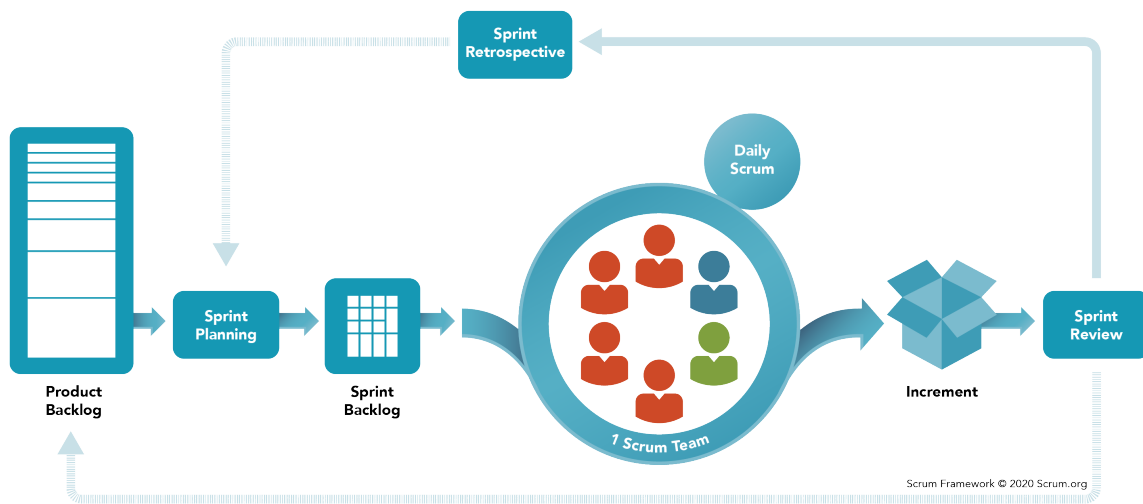


Figura 3.1: Marco de Trabajo Scrum

### 3.1.1. Gestión del Esfuerzo

Dentro de este apartado se pretende mostrar una relación entre los objetivos planificados en cada uno de los Sprints del proyecto y el esfuerzo real dedicado (en horas de trabajo) a la consecución de dichos objetivos. Ser conscientes del esfuerzo que nos requiere conseguir ciertos objetivos, nos hará mejorar a la hora de realizar futuras estimaciones de objetivos o tareas análogas de una forma más precisa.

En concreto, este proyecto se ha organizado en 5 Sprints de 3 semanas de duración cada uno. En la planificación de inicio de cada Sprint fueron propuestos unos objetivos de alto nivel a cumplir durante el transcurso Sprint.

A continuación se presenta una tabla de Gestión del Esfuerzo en donde se reflejan los objetivos planteados en la planificación de cada Sprint del proyecto junto con el esfuerzo empleado para su consecución (medido en horas) y su grado de cumplimiento.

SPRINT	OBJETIVOS ASOCIADOS	ESFUERZO (HORAS)
#1	<ul style="list-style-type: none"> <li>✓ Asimilación de conceptos y tecnologías involucradas</li> <li>✓ Estudio del estado del arte</li> </ul>	60 h
#2	<ul style="list-style-type: none"> <li>✓ Identificación de objetivos del proyecto</li> <li>✓ Comienzo de la implementación de la herramienta</li> </ul>	49 h
#3	<ul style="list-style-type: none"> <li>✓ Implementación de una versión funcional de la herramienta</li> </ul>	69 h
#4	<ul style="list-style-type: none"> <li>✓ Finalización de la Interfaz de Usuario</li> <li>✗ Integración del modelo de asistencia a la anotación</li> </ul>	70 h
#5	<ul style="list-style-type: none"> <li>✓ Integración del modelo de asistencia a la anotación</li> <li>✓ Procesamiento OCR de la entrada</li> </ul>	108 h

Figura 3.2: Gestión del Esfuerzo

*Nota:* de acuerdo a las bases de Scrum, solo se consideran cumplidos los objetivos cuyas tareas asociadas fueron terminadas al 100 %.

## 3.2. Estimación

En este apartado se llevará a cabo una estimación del esfuerzo necesario que se presupone para realizar cada una de las tareas planificadas en el proyecto. A partir de los objetivos del proyecto, se plantean una serie de historias de usuario con las que se pretende abarcar dichos objetivos para su cumplimiento.

Las principales historias de usuario comprendidas en este proyecto se listan a continuación.

### ■ HU-01: ESTUDIO DE LAS HERRAMIENTAS DE ANOTACIÓN EXISTENTES

Consiste en el descubrimiento e investigación del ecosistema de herramientas de anotación existentes y la selección de varias de las más utilizadas para la realización de un posterior estudio de características ofrecidas y funcionamiento general a partir del cual poder sacar unas conclusiones de utilidad con vistas a la futura construcción de nuestra propia herramienta.

### ■ HU-02: IMPLEMENTACIÓN DE LA APLICACIÓN WEB DE ANOTACIÓN

Consiste en el proceso de implementación de la interfaz de usuario y lógica de negocio necesarias para proveer a la aplicación del funcionamiento básico esperado en una herramienta de anotación. Entre las características fundamentales a implementar se encuentran la capacidad para cargar nuevos textos a la aplicación, anotar entidades categorizadas dentro de los textos y exportar los textos anotados. Aparte de esta gestión de documentos,

anotaciones y categorías de entidad, también se encuadran dentro de esta historia otras características más específicas como pueden ser la visualización de estadísticas de anotación o la gestión de expresiones regulares vinculadas a categorías de entidad.

#### ■ **HU-03: CAPACIDAD DE ASISTENCIA A LA ANOTACIÓN DE ENTIDADES**

Consiste en la habilitación de modos de asistencia a la anotación con los que agilizar el proceso completo de anotación por parte del usuario de la herramienta. Dentro de los modos de asistencia se incluyen el uso de patrones de coincidencia basados en expresiones regulares vinculadas a categorías de entidades, el entrenamiento de modelos de Reconocimiento de Entidades Nombradas a partir de las anotaciones existentes en un dataset de textos y la predicción utilizando modelos de Reconocimiento de Entidades Nombradas preentrenados e importados a la herramienta.

#### ■ **HU-04: PROCESO DE DIGITALIZACIÓN DE DOCUMENTOS**

Consiste en el proceso de carga y procesamiento OCR (Reconocimiento Óptico de Caracteres) del fichero de entrada para la derivación de una representación textual de su contenido que pueda ser manejable por un ordenador como una cadena de texto plano. Posteriormente, el texto obtenido deberá ser integrado dentro del flujo de proceso de la herramienta para proceder a su anotación.

#### ■ **HU-05: REDACCIÓN DE DOCUMENTACIÓN DEL PROYECTO**

Consiste en la redacción de la presente memoria del proyecto, abarcando los diferentes capítulos y secciones que la componen.

A su vez, estas historias de usuario se dividirán en tareas más pequeñas que serán las abordadas durante los Sprints para lograr los objetivos del proyecto.

Antes de planificar las tareas en las que trabajar en un Sprint, se debe hacer una estimación del esfuerzo requerido para completarlas. Esta estimación debe llevarse a cabo mediante alguna técnica de estimación. En nuestro caso, la técnica escogida es Planning Poker [18], en la que se asigna a cada tarea una dificultad en Puntos de Historia, siguiendo una escala de valores determinada por la Serie de Fibonacci. Cabe destacar que la serie de Fibonacci es aquella en la que cada número que la compone se obtiene realizando la suma de los 2 números inmediatamente anteriores. Esta escala de valores es incremental de manera que a mayor dificultad estimada para la tarea, mayor valor de Puntos de Historia serán asignados a la misma.

A continuación, se procede a especificar las tareas asociadas a cada historia de usuario junto con la estimación del esfuerzo requerido para completarlas.

<b>ID Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación (Puntos de Historia)</b>
T-01.01	Investigación del conjunto de herramientas de anotación existentes en la actualidad.	2
T-01.02	Análisis del funcionamiento y características de las herramientas de anotación seleccionadas.	5
T-01.03	Comparativa de las características presentes en las herramientas de anotación analizadas.	3
T-01.04	Obtención de conclusiones finales del estudio y derivación de oportunidades para nuestra herramienta de anotación.	2

Tabla 3.1: Tareas asociadas a la Historia de Usuario HU-01

<b>ID Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación (Puntos de Historia)</b>
T-02.01	Implementación de la gestión de anotaciones en un texto.	8
T-02.02	Implementación de la gestión de categorías de entidad.	5
T-02.03	Implementación de la importación de nuevos documentos.	3
T-02.04	Implementación de la navegabilidad entre los documentos.	2
T-02.05	Implementación de la navegabilidad entre las sentencias de un documento.	2
T-02.06	Implementación de la gestión de expresiones regulares.	3
T-02.07	Implementación de la asociación entre expresiones regulares y categorías de entidad.	2
T-02.08	Implementación de la gestión de proyectos.	5
T-02.09	Implementación de la exportación del dataset anotado.	3
T-02.10	Implementación de la importación de un dataset anotado.	5
T-02.11	Implementación de la exportación de fichero de expresiones regulares.	2
T-02.12	Implementación de la importación y carga de fichero de expresiones regulares.	3
T-02.13	Implementación de la visualización de las estadísticas de anotación.	2
T-02.14	Conexión del frontend y el backend de la aplicación para su efectiva comunicación.	8

Tabla 3.2: Tareas asociadas a la Historia de Usuario HU-02



<b>ID Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación (Puntos de Historia)</b>
T-03.01	Implementación de la asistencia a la anotación mediante el uso de patrones de coincidencia.	5
T-03.02	Implementación de la asistencia a la anotación mediante el entrenamiento de un modelo de aprendizaje automático y el uso de patrones de coincidencia.	8
T-03.03	Implementación de la asistencia a la anotación por medio de las predicciones del modelo de aprendizaje automático importado al proyecto.	3
T-03.04	Implementación de la exportación del modelo de aprendizaje automático entrenado.	2
T-03.05	Implementación de la importación y carga de un modelo de aprendizaje automático preentrenado al proyecto.	2
T-03.06	Implementación de la visualización de resultados del entrenamiento de un modelo aprendizaje automático.	1

Tabla 3.3: Tareas asociadas a la Historia de Usuario HU-03

<b>ID Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación (Puntos de Historia)</b>
T-04.01	Implementación de la importación de documentos PDF.	2
T-04.02	Implementación del procesamiento OCR sobre el documento de entrada para la obtención del texto contenido.	8

Tabla 3.4: Tareas asociadas a la Historia de Usuario HU-04

<b>ID Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación (Puntos de Historia)</b>
T-05.01	Redacción del capítulo de Introducción.	5
T-05.02	Redacción del capítulo de Estado del Arte.	3
T-05.03	Redacción del capítulo de Planificación y Presupuesto.	5
T-05.04	Redacción del capítulo de Análisis.	5
T-05.05	Redacción del capítulo de Diseño.	5
T-05.06	Redacción del capítulo de Implementación.	5
T-05.07	Redacción del capítulo de Conclusiones y Trabajo Futuro.	3
T-05.08	Redacción de la sección de Manuales.	2
T-05.09	Revisión general de la memoria.	5

Tabla 3.5: Tareas asociadas a la Historia de Usuario HU-05

En total, se ha estimado el alcance propuesto para el proyecto en 134 puntos de historia, cuya realización se llevará a cabo en los Sprints que componen el proyecto.

### 3.3. Planificación Temporal

Una vez especificadas las tareas que abarcarán el alcance de nuestro proyecto, se procede a realizar la planificación temporal de las mismas dentro de cada uno de los Sprints de nuestro proyecto.

La asignatura Trabajo Fin de Grado (TFG) lleva asociado un peso de 12 créditos ECTS, lo que equivale a 300 horas de trabajo. Por tanto, esta restricción se debe tener en cuenta a la hora de planificar tanto el alcance de la herramienta como la organización del trabajo a realizar en los Sprints de los que se compone el proyecto.

La planificación temporal relativa a este proyecto se ha organizado en torno a 5 Sprints de una duración de 3 semanas cada uno, empezando el día 10/03/2021 y terminando el día 30/06/2021. Debido a la diferencia en mi disponibilidad de horas para realizar el TFG durante los Sprints, el alcance planificado para cada uno ha ido adaptándose a estas condiciones. Mientras que en los Sprints 1-3 la dedicación de tiempo al TFG fue parcial, en los Sprints 4 y 5 ésta ya fue completa. Por ello, en en los últimos Sprints del proyecto ha podido dedicarse un mayor esfuerzo temporal que en el resto.

<b>Sprint</b>	<b>Intervalo de Tiempo</b>	<b>Tareas Asignadas</b>	<b>Esfuerzo (Puntos de Historia)</b>
#1	10/03/2021 - 07/04/2021	T-01.01, T-01.02, T-01.03, T-01.04, T-05.02, T-05.03	20
#2	07/04/2021 - 28/04/2021	T-02.01, T-02.02, T-02.03, T-05.01, T-05.04	23
#3	28/04/2021 - 19/05/2021	T-02.04, T-02.05, T-02.06, T-02.07, T-02.11, T-02.12, T-05.04, T-05.05	22
#4	19/05/2021 - 09/06/2021	T-02.08, T-02.13, T-02.14, T-03.01, T-03.02, T-03.03, T-03.04, T-03.05, T-03.06	36
#5	09/06/2021 - 30/06/2021	T-02.09, T-02.10, T-04.01, T-04.02, T-05.06, T-05.07, T-05.08, T-05.09	33

Tabla 3.6: Planificación Temporal Inicial de los Sprints

Dicha planificación temporal por Sprints también aparece ilustrada visualmente en el Diagrama de Gantt de la Figura 3.3.

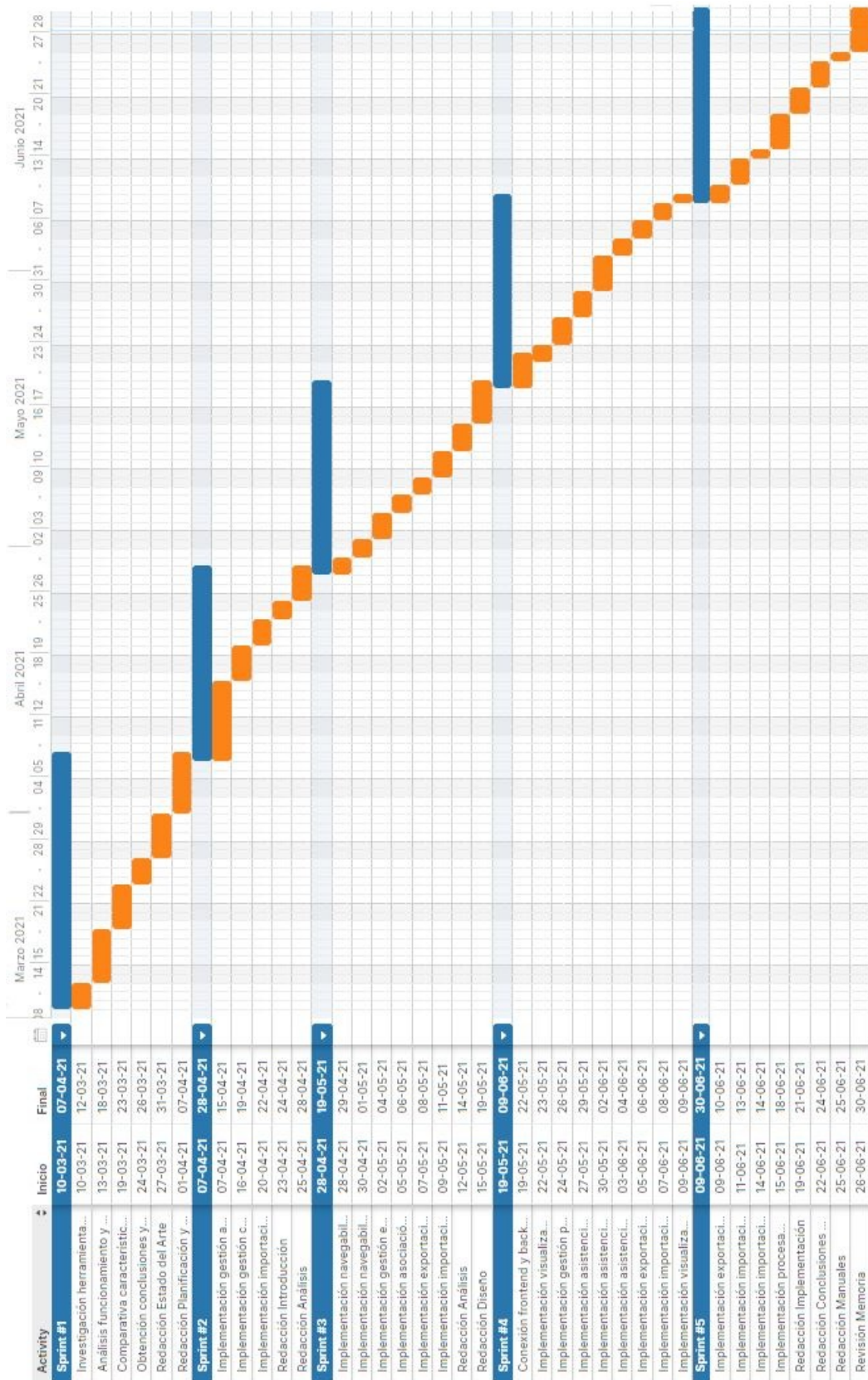


Figura 3.3: Diagrama de Gantt Inicial

### Planificación Temporal Final

A lo largo del transcurso del proyecto se tuvo que replanificar la propuesta inicial debido a algunas variaciones acontecidas. Concretamente, durante el Sprint 4 no se pudo completar ninguna de las tareas asociadas a la Historia de Usuario **Capacidad de Asistencia a la Anotación de Entidades** planteadas dentro del alcance del Sprint, por lo que se vio necesario mover dichas tareas restantes al Sprint siguiente. El hecho de no poder implementar todas las tareas planificadas dentro del alcance del Sprint 4 hace indicar que posiblemente subestimamos el esfuerzo que sería necesario para completarlas.

Una vez comentados estos cambios, en la siguiente tabla se muestra la planificación temporal final asumida.

Sprint	Intervalo de Tiempo	Tareas Asignadas	Esfuerzo (Puntos de Historia)
#1	10/03/2021 - 07/04/2021	T-01.01, T-01.02, T-01.03, T-01.04, T-05.02, T-05.03	20
#2	07/04/2021 - 28/04/2021	T-02.01, T-02.02, T-02.03, T-05.01, T-05.04	23
#3	28/04/2021 - 19/05/2021	T-02.04, T-02.05, T-02.06, T-02.07, T-02.11, T-02.12, T-05.01, T-05.05	21
#4	19/05/2021 - 09/06/2021	T-02.08, T-02.13, T-02.14	15
#5	09/06/2021 - 30/06/2021	T-02.09, T-02.10, T-03.01, T-03.02, T-03.03, T-03.04, T-03.05, T-03.06, T-04.01, T-04.02, T-05.06, T-05.07, T-05.08, T-05.09	54

Tabla 3.7: Planificación Temporal Final de los Sprints

*Nota:* entre los cambios destaca el paso al Sprint 5 de las tareas asociadas a la Historia de Usuario **Capacidad de Asistencia a la Anotación de Entidades**.

Igualmente, se muestra el diagrama de Gantt actualizado con la planificación temporal final del proyecto en la Figura 3.4.

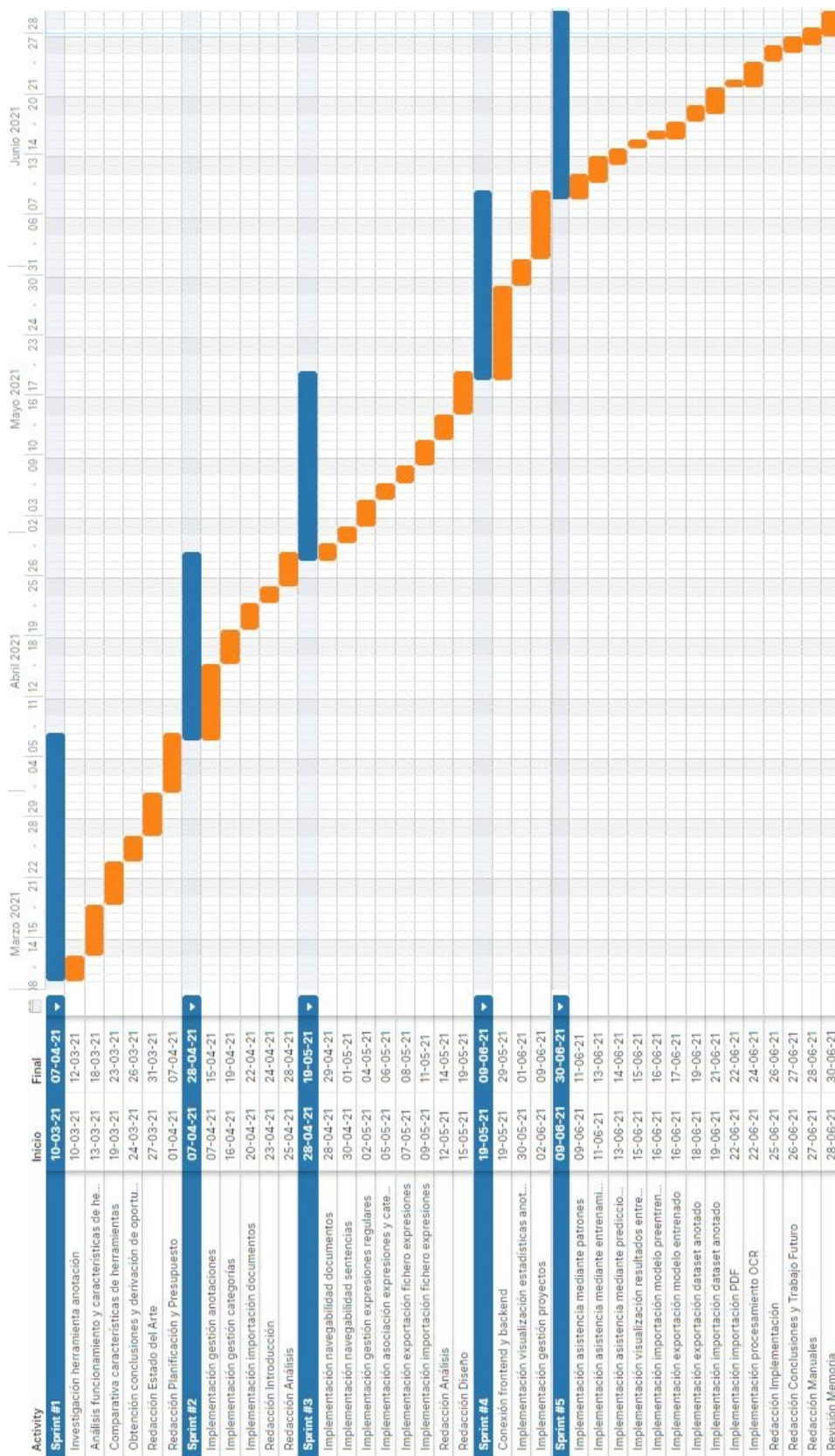


Figura 3.4: Diagrama de Gantt Final

### 3.4. Presupuesto

El siguiente paso tras completar la planificación temporal de nuestro proyecto será llevar a cabo el cálculo de su presupuesto. El presupuesto es una estimación económica del coste que prevemos que conllevará la realización de un proyecto. El hecho de ser una estimación implica que el coste final real al concluir el proyecto puede variar con respecto a lo previsto inicialmente.

Para la determinación del presupuesto de un proyecto software tendremos en cuenta una serie de costes asociados a varios aspectos entre los que se enumeran el hardware, el software y los recursos humanos.

Desde el punto de vista del hardware, se ha trabajado con un ordenador portátil de prestaciones medias, dentro de las cuales destacan las siguientes.

Componente	Prestación
Procesador	Intel Core i7 - 7ª Generación
Tarjeta Gráfica	AMD Radeon R7 M440
Disco Duro	480 GB de SSD
Memoria RAM	8 GB

Tabla 3.8: Prestaciones Ordenador Portátil de Desarrollo

Su coste de adquisición junto con el correspondiente prorrateo acorde al tiempo de dedicación al proyecto se muestran en la siguiente tabla.

Ítem	Precio	Tiempo de Vida Útil	Tiempo Proyecto	Uso en el proyecto	Prorrateo / Coste
Ordenador Portátil	650 €	5 años (1826 días)	105 días	$105 * 100 / 1826 =$ 5,75 %	$650 * 105 / 1826 =$ <b>37,38 €</b>
<b>TOTAL</b>	<b>37,38 €</b>				

Tabla 3.9: Costes de Hardware

Dentro de los costes de Software se incluyen principalmente los costes asociados licencias de sistema operativo, programas o herramientas requeridas. A pesar de estar instalado en el equipo el Sistema Operativo Windows 10, el desarrollo se llevó a cabo sobre una Máquina Virtual con la distribución Ubuntu de Linux, por lo que dentro de estos costes no consideraremos la propia licencia de Windows 10. A su vez, al desarrollar sobre un sistema operativo de código abierto, como es Linux, la totalidad de las licencias de software de programas y herramientas utilizadas son gratuitas; por lo que no repercuten sobre el presupuesto.

Ítem	Licencia	Coste
Microsoft Teams	Gratuita	0 €
Trello	Libre	0 €
Visual Studio Code	Libre	0 €
VirtualBox	Libre	0 €
Anaconda	Libre	0 €
TeXstudio	Libre	0 €
StarUML	Libre	0 €
<b>TOTAL</b>		<b>0 €</b>

Tabla 3.10: Coste de Software

A continuación, acorde a la planificación temporal planteada inicialmente se calculan los costes relativos a los recursos humanos necesarios para desarrollar todo el proyecto. Para ello, se deben tener en cuenta los diferentes roles de profesionales implicados.

En lo que a este proyecto se refiere, se consideran necesarios los roles de Jefe de Proyecto, Analista, Desarrollador Frontend y Desarrollador Backend. Para la estimación del salario de cada uno de los 4 tipos de profesionales, se ha recurrido al sitio web de LinkedIn Salary <sup>1</sup>, el cual proporciona los datos acerca del salario medio según el puesto desempeñado.

En la siguiente tabla se muestran los costes asociados al trabajo de los diferentes tipos de recursos humanos, estimando las horas requeridas por cada uno para la realización del proyecto.

Recurso	Salario	Horas	Coste
Jefe de Proyecto	3500 € / mes (20,80 € / hora)	48	20,80*48 = 998,40 €
Analista de Software	2600 € / mes (15,50 € / hora)	72	15,50*72 = 1116 €
Desarrollador Frontend	2080 € / mes (12,40 € / hora)	90	12,40*90 = 1116 €
Desarrollador Backend	2250 € / mes (13,40 € / hora)	90	13,40*90 = 1206 €
<b>TOTAL</b>			<b>4436,40 €</b>

Tabla 3.11: Coste de Recursos Humanos

<sup>1</sup><https://www.linkedin.com/salary/>



*Nota:* para la obtención del Salario Diario se ha considerado una jornada laboral estándar de 21 días laborables y 8 horas diarias.

Por otra parte, dentro de Otros Costes se incluyen generalmente los costes asociados a la Conexión a Internet, costes de electricidad o material de oficina. En este proyecto, los Otros Costes se limitan a la necesidad de Conexión a Internet y la electricidad, tal y como se muestran en la siguiente tabla.

Ítem	Coste Mensual	Tiempo Proyecto	Prorrateo / Coste
Conexión a Internet	50 €	105 días (3,5 meses)	$3,5 * 50 = 175$ €
Electricidad	10 €	105 días (3,5 meses)	$3,5 * 10 = 35$ €
<b>TOTAL</b>	<b>210 €</b>		

Tabla 3.12: Otros Costes

Una vez calculados los diferentes tipos de costes, ya podemos obtener el presupuesto económico total.

Recurso	Coste
Hardware	37,38 €
Software	0 €
Humanos	4436,40 €
Otros	210 €
<b>TOTAL</b>	<b>4683,78 €</b>

Tabla 3.13: Presupuesto Inicial

### Presupuesto Final

En la sección anterior vimos que no pudieron ser realizadas varias tareas de backend propuestas dentro del alcance del Sprint 4, lo cual causó que dichas tareas fueras replanificadas para su implementación en el siguiente Sprint.

Debido a que en el Sprint 5 se pudieron satisfacer las tareas asociadas a los nuevos objetivos propuestos en su alcance no fue necesario ampliar el periodo de tiempo del proyecto con la inclusión de un nuevo Sprint. Lo que sí se vio incrementado fue el tiempo dedicado al proyecto, superando las 300 horas previstas inicialmente. Finalmente, el esfuerzo requerido para la realización del proyecto llegó a las 348 horas. Debido a ello, se ve necesario incrementar el número de horas trabajadas por el Desarrollador de Backend.

En definitiva, el coste final del proyecto variará con respecto al planteado inicialmente debido al aumento de horas empleadas.

La siguiente tabla presenta los costes de recursos humanos actualizados según los ajustes requeridos.

<b>Recurso</b>	<b>Salario</b>	<b>Horas</b>	<b>Coste</b>
Jefe de Proyecto	3500 € / mes (20,80 € / hora)	48	20,80*48 = 998,40 €
Analista de Software	2600 € / mes (15,50 € / hora)	72	15,50*72 = 1116 €
Desarrollador Frontend	2080 € / mes (12,40 € / hora)	90	12,40*90 = 1116 €
Desarrollador Backend	2250 € / mes (13,40 € / hora)	138	13,40*138 = 1849,20 €
<b>TOTAL</b>	<b>5079,60 €</b>		

Tabla 3.14: Coste de Recursos Humanos Actualizado

*Nota:* solo varían los costes asociados a las horas adicionales trabajadas por el recurso Desarrollador Backend.

Finalmente, el presupuesto final queda de la siguiente forma.

<b>Recurso</b>	<b>Coste</b>
Hardware	37,38 €
Software	0 €
Humanos	5079,60 €
Otros	210 €
<b>TOTAL</b>	<b>5326,98 €</b>

Tabla 3.15: Presupuesto Final

Por lo tanto, el coste final establecido del proyecto sería de 5326,98 €, lo que supone una variación de 643,20 € más respecto a la estimación del presupuesto inicial.

# Capítulo 4

## Fundamentos Teóricos

En este capítulo se describen los fundamentos teóricos relativos a las principales tecnologías y componentes del dominio del proyecto. Se busca dar una primera aproximación acerca de los conceptos más importantes involucrados en el uso y desarrollo de la herramienta.

A continuación, se proceden a explicar en detalle cada uno de estos componentes del dominio.

### 4.1. Anotación

La anotación es el proceso de etiquetado de entidades en elementos de diverso tipo, principalmente textos e imágenes, de manera que quede representada la información relevante encontrada en ellos.

Debido a que este proyecto se centra en las documentos de tipo textual, referiremos la anotación de entidades como el proceso de etiquetado de las entidades nombradas en un texto con el fin de clasificar su información más relevante. Dentro de los tipos de entidades más comunes que suelen anotarse están los nombres de personas, organizaciones, localizaciones geográficas, marcas temporales, emails, DNIs, URLs, etc. No obstante, pueden considerarse otros tipos de entidades distintos definidos por el propio usuario.

Dentro del ámbito de una herramienta de anotación con interfaz gráfica, las entidades de un texto puedan ser etiquetadas por medio del uso de ratón, así como modificadas o eliminadas a través de alguna opción en la interfaz o del propio uso de ratón.

Librerías de procesamiento del lenguaje natural, como Spacy o NLTK (Natural Language Toolkit), proveen modelos de aprendizaje preentrenados y métodos para obtener las entidades nombradas más típicas con las que poder anotar un texto recibido como entrada. Estas librerías se componen de pipelines de procesamiento específicos preparados para ser utilizados en diferentes contextos según el lenguaje o ámbito en el que esté escrito el documento de texto sobre el que se desea procesar el pipeline. Así mismo, según la tarea de Procesamiento del Lenguaje Natural que se desee llevar a cabo, se podrán elegir pipelines configurados con los componentes más apropiados.

Por otra parte, dentro del proceso de anotación se incluye el modo de representación utilizado para almacenar las entidades identificadas en un texto. En el caso de especificar las entidades

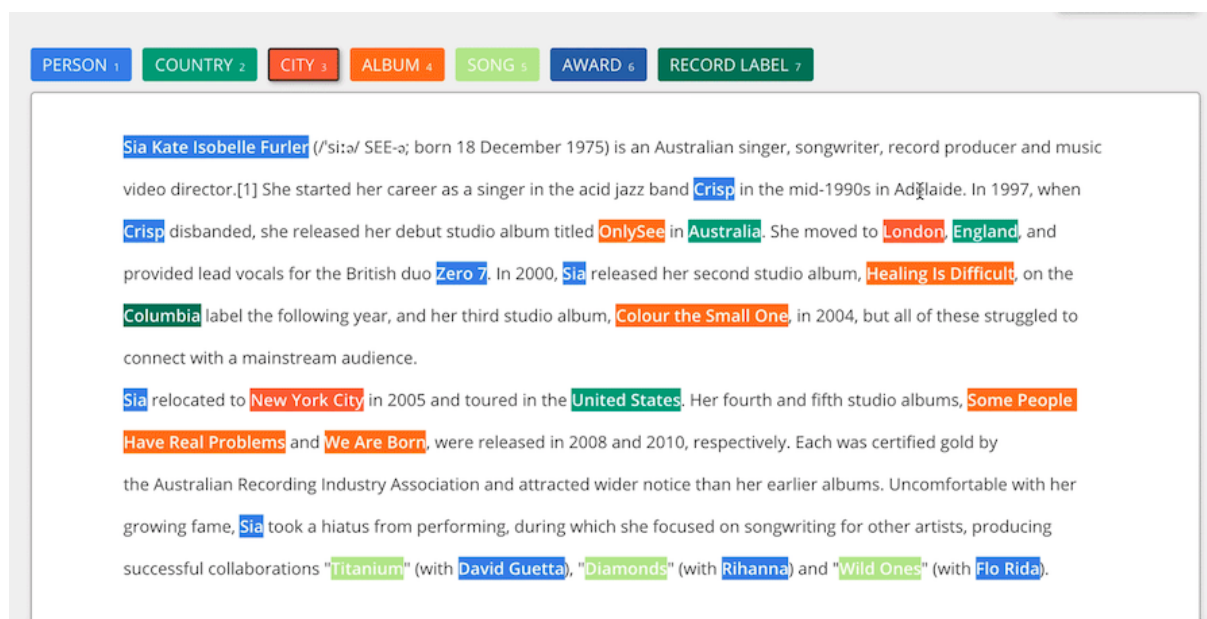


Figura 4.1: Interfaz Gráfica de Anotación de Entidades

dentro de un fichero, el esquema de anotación más popular es el formato IOB, junto con sus principales variantes. Este formato de anotación, se basa en el uso de 3 etiquetas (I, O y B) para asociar con cada token (palabras, signos de puntuación o espacios) de un texto.

Entre las variantes más populares del esquema IOB destacan el formato IO (que prescinde de la etiqueta B y marca con la etiqueta I al primer token de una entidad) o el formato BI que prescinde de las etiquetas de tipo O, solo estableciendo las correspondientes a tokens dentro de entidades. También, en las soluciones basadas en ficheros es común utilizar un formato JSON debido a su portabilidad y facilidad de lectura.

Otras opciones de almacenamiento de las entidades anotadas en un texto pasan por el uso de una base de datos. Dentro de este enfoque, es común emplear una base de datos de tipo documental para guardar tanto el texto y sus anotaciones como metadatos concretos acerca del mismo, tales como la fuente de obtención del texto o su idioma.

El uso de una solución basada en ficheros o una base de datos dependerá del caso de uso concreto.

## 4.2. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) es un campo dentro de la Inteligencia Artificial y la Lingüística que trata el estudio del lenguaje natural para la construcción de modelos de lenguaje entendibles por los ordenadores. Su objetivo consiste en transformar la información lingüística usada por las personas para comunicarse en una representación de dicha información adecuada para ser procesada por un ordenador.

El estudio en Procesamiento del Lenguaje Natural (PLN) surgió en la época de los años 1950s

con trabajos como el test de Turing (un examen de la capacidad de un ordenador para exhibir la inteligencia de los humanos) o el experimento Georgetown-IBM (un intento de traducción del idioma ruso al inglés). Años más tarde, a finales de la década de 1980s, el estudio relativo al procesamiento natural tomó un nuevo rumbo gracias al auge de algoritmos eficientes de aprendizaje automático, estableciéndose el uso de modelos estadísticos como principal aproximación en el área del PLN y dejando atrás la anterior aproximación basada en la lógica simbólica. Así, el procesamiento del lenguaje natural resurgió acompañado de mejores resultados hasta la actualidad, en la que la aplicación de algoritmos de deep learning (aprendizaje profundo) está llevando a resultados aún más prometedores. No obstante, se sigue teniendo margen de mejora pues el procesamiento del lenguaje natural es un campo con numerosas dificultades y muy dependiente de las características del idioma sobre el que se aplique.

El proceso de la tecnología de Procesamiento del Lenguaje Natural consta de una serie de etapas que variarán dependiendo del tarea concreta con la que se trabaje dentro de este campo. Algunas de las modalidades más destacadas dentro del Procesamiento del Lenguaje Natural son las siguientes:

- Extracción de Información
- Reconocimiento del Habla
- Generación de Lenguaje Natural
- Detección de Sentimientos
- Clasificación de textos
- Traducción automática

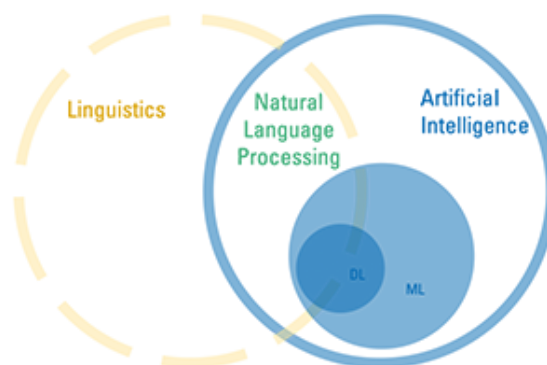


Figura 4.2: Situación Procesamiento del Lenguaje Natural

Concretamente, en este proyecto nos centraremos en el proceso llevado a cabo para el Reconocimiento de Entidades Nombradas (NER), el cual es a su vez un subcampo dentro de la Extracción de Información.

### 4.3. Reconocimiento de Entidades Nombradas

El Reconocimiento de Entidades Nombradas (NER) es una tarea de Extracción de Información dentro del Procesamiento del Lenguaje Natural que consiste en la identificación y clasificación en categorías predefinidas de las entidades nombradas existentes en documentos de texto. Las categorías en las que se clasifican las partes de un texto pueden ser muy diversas, como:

- Nombres de personas
- Nombres de organizaciones
- Localizaciones geográficas
- Expresiones temporales
- Cantidades monetarias
- Direcciones de correo electrónico
- URLs
- DNIs

Estas categorías también pueden ser creadas y descritas por un potencial usuario, de modo que se permita categorizar información personalizada a medida de las necesidades del mismo. El objetivo es construir un modelo de Reconocimiento de Entidades Nombradas (NER) que permita hacer predicciones de entidades en documentos de texto nuevos, de manera que se pueda extraer información relevante de forma automática.

Para llevar a cabo el proceso de Reconocimiento de Entidades Nombradas es preciso disponer de conjuntos de datos etiquetados con ejemplos de textos y sus entidades anotadas, identificadas previamente por personas u otros sistemas cuya eficacia haya sido probada. Además, es recomendable que dicho conjunto de datos de entrenamiento sea grande para así poder construir un modelo de NER lo más eficiente posible. Cuantos más ejemplos de entrenamiento empleemos durante la fase de entrenamiento, menos riesgo de sobreaprendizaje tendremos y mejor será el modelo que construyamos prediciendo nuevas entidades en textos nunca vistos.

Para derivar el resultado correspondiente a la tarea de Reconocimiento de Entidades Nombradas se debe seguir un proceso de varias etapas en las que procesar el documento de texto recibido y predecir sus entidades asociadas. Un pipeline de procesamiento para el Reconocimiento de Entidades Nombradas puede estar formado por distintos tipos de componentes que contribuyan a un mejor desempeño en la realización de esta tarea. Una aproximación comúnmente utilizada implica dividir el proceso en 2 partes diferenciadas:

## 1. Preprocesamiento del texto

## 2. Reconocimiento de Entidades Nombradas en el texto

Dentro de la etapa de Preprocesamiento del texto se incluyen los siguientes subprocesos:

### ■ Preprocesamiento

1. **Tokenización:** este subproceso consiste en la división del texto en tokens de manera que se obtengan las partes elementales del texto.

Esta división debe tener en cuenta las características, patrones y normas léxicas del idioma en el que está escrito el texto. Además, será acorde al lenguaje en el que está escrito, teniendo en cuenta aspectos como contracciones o abreviaturas, y no meramente partiendo el texto por sus espacios en blanco.

2. **Segmentación de sentencias:** este paso trata la agrupación de los tokens para construir las sentencias que componen el texto.

Este proceso se lleva a cabo determinando los límites de las sentencias teniendo en cuenta principalmente los signos de puntuación y saltos de línea.

3. **Etiquetado gramatical (Part-of-speech tagging):** este proceso consiste en la clasificación de los tokens del texto dentro de categorías gramaticales como nombres, verbos, adjetivos, adverbios, determinantes... Para ello, se etiqueta cada token del texto con la clase gramatical que le corresponde.

Esta tarea puede requerir un análisis morfológico del texto que incluye otros subprocesos específicos como la reducción, la lematización o la desambiguación de ciertas partes del mismo.

- **Reducción:** consiste en la determinación de la forma base reducida de la que deriva una palabra.  
Ejemplos: niños → niñ, formalidad → formal
- **Lematización:** consiste en la derivación del lema asociado a la forma flexionada de una palabra.  
Ejemplos: soy → ser, jugando → jugar
- **Desambiguación:** consiste en la identificación de la clase gramatical que tiene una determinada palabra según el contexto concreto en el que se encuentra.  
Comúnmente, el problema de la ambigüedad sucede cuando una palabra puede actuar como nombre, verbo o adjetivo dependiendo del contexto en el que esté.

4. **Determinación de la estructura gramatical (Phrase structure):** este proceso consiste en la determinación de la estructura gramatical de las sentencias del texto a través de la asignación de las relaciones gramaticales entre sus partes.

Estas tareas de preprocesamiento del texto tienen como objetivo mejorar la eficacia del proceso de Reconocimiento de Entidades Nombradas. Primero, el componente dedicado al Reconocimiento de Entidades Nombradas recibirá como entrada el texto preprocesado por las tareas

anteriores. Posteriormente, a partir del texto enriquecido se podrá efectuar el reconocimiento de las entidades nombradas mediante el uso de patrones de coincidencia y/o modelos de aprendizaje.

A continuación, se describe el funcionamiento de la etapa asociada a la propia tarea de Reconocimiento de Entidades Nombradas.

### ■ Reconocimiento de Entidades Nombradas:

Dentro de esta etapa principal se incluye la predicción de las entidades nombradas que hay en el texto por parte de un modelo de NER. Para la implementación de un modelo de NER existen 2 grandes aproximaciones, las cuales puede ser combinadas para obtener resultados más precisos y mejorar la eficacia del modelo resultante.

- **Patrones de coincidencia (Pattern matching):** los patrones son reglas que se aplican sobre un texto para obtener las partes del mismo que coinciden con su definición. Éstos pueden ser útiles cuando las entidades buscadas en el texto tienen una estructura definida o un número no muy extenso de instancias posibles. Por ejemplo, el uso de estos patrones podría funcionar bien con entidades restringidas como direcciones de correo electrónico, URLs, DNIs o nombres de países, entre otros.

Con este enfoque no es necesario aplicar ningún modelo de aprendizaje para hallar las entidades deseadas, tan solo se necesitan definir buenos patrones que permitan identificar las partes del texto involucradas. Con librerías de PLN como Spacy se pueden establecer, a nivel programático, patrones para entidades cuyas instancias se ajusten a unos criterios definidos a nivel de token o secuencia de tokens, siendo un token la representación de una parte del texto ya sea una palabra, un símbolo de puntuación o un espacio en blanco. Se pueden especificar atributos de distinto tipo para hacer coincidir con los tokens del texto. Éstos abarcan desde buscar un token por su escritura en minúsculas (lowercase) hasta utilizar elaboradas expresiones regulares para ello. En todo caso, estos patrones son definidos por el desarrollador y pueden ser tan complejos como se requieran.

- **Modelo de aprendizaje:** por otra parte tenemos los modelos de aprendizaje, cuya implementación se basa en el entrenamiento a partir de extensos datasets con textos de ejemplo junto con sus entidades anotadas. Es decir, se proporcionan grandes cantidades de textos etiquetados con sus entidades nombradas deseadas.

En la mayoría de los casos, un modelo de aprendizaje permitirá obtener mejores resultados y predecir entidades en base al contexto, que con patrones de coincidencia nunca obtendríamos. Además, gracias a dicho contexto, un modelo de aprendizaje podría reconocer entidades mal escritas en el texto que realmente sí interesa clasificar. Esta característica es especialmente relevante en nuestro proyecto, ya que parte de los textos empleados provendrán de documentos digitalizados por medio de un motor OCR, el cual puede llegar a cometer errores en el proceso de conversión de algún carácter.



La solución más óptima pasa por combinar ambos enfoques de manera que los patrones de coincidencia puedan reconocer fácilmente entidades con estructura definida y pasárselas al modelo de aprendizaje como parte de su entrenamiento para así agilizar el proceso.

Para otras tareas de Procesamiento del Lenguaje Natural se deberá seguir otro pipeline de procesamiento diferente, acorde a las necesidades que requiera la tarea en curso.

### 4.4. Reconocimiento Óptico de Caracteres

El Reconocimiento Óptico de Caracteres (OCR) es una tecnología utilizada para la digitalización de documentos de texto cuyo funcionamiento se basa en la transformación de una imagen en 2 dimensiones con texto incluido en una representación del texto manejable por un ordenador. Para ello, el proceso que sigue incluye la detección y reconocimiento de los caracteres presentes en el fichero recibido como entrada al motor OCR.

La tecnología OCR ha tomado una gran importancia a raíz del proceso de transformación digital actual, empleándose esencialmente en los 2 siguientes casos de uso.

- **Fotografías:** la derivación del texto en imágenes tomadas en medios no restringidos (como la naturaleza) puede ser un proceso complejo dependiente del entorno en el que se tomó la imagen o la calidad del dispositivo con la que se tomó. Existen muchos factores que pueden influir en la correcta visibilidad del texto de una imagen por un motor OCR, tales como su orientación, el color de fondo, la saturación, si la imagen está borrosa...

Debido a ello, se requiere el uso de filtros de preprocesamiento para aplicar sobre las imágenes y poder clarificar el texto contenido en ellas. Una librería de Visión Computacional puede ser utilizada a estos efectos.

Ejemplos de uso: matrículas de vehículos, señales de tráfico, lecturas de contadores...

- **Documentos escaneados:** este tipo de documentos poseen generalmente una estructura más definida y la derivación del texto en ellos suele ser más sencilla. El fondo de la imagen suele ser más homogéneo y la calidad mejor si es escaneada con un buen dispositivo.

No obstante, dentro de esta categoría también se incluyen los textos manuscritos que pueden requerir un esfuerzo adicional para derivar el texto correctamente, pues puede ser necesario el entrenamiento con datasets de caracteres muy variados y extensos para derivar un buen resultado.

Ejemplos de uso: facturas, pasaportes, cartas de restaurantes...

En estos años, los motores OCR se han servido de los últimos avances en el área de aprendizaje profundo (Deep Learning) para mejorar la eficacia de sus resultados. En concreto, las redes LSTM han supuesto una mejora considerable para inferir de forma efectiva el texto que se encuentra en un fichero de entrada que se suministra al motor OCR. El funcionamiento de este tipo de redes neuronales permite reconocer dependencias en el texto a largo plazo, de modo que se puedan intuir partes del texto a partir de otras anteriores.

A continuación, se describe el proceso que sigue un motor OCR desde que recibe el fichero de entrada hasta que devuelve el resultado obtenido. Este proceso consiste en una serie de subprocesos necesarios para obtener un resultado lo más acertado posible.

- **Preprocesamiento de la imagen:** esta primera etapa es altamente recomendable en la mayoría de las situaciones para obtener un resultado lo más acertado posibles.

El proceso consiste en aplicar una serie de filtros y transformaciones sobre la imagen recibida como entrada para adecuarla un formato que mejore el desempeño de los procesos subsiguientes que se llevarán a cabo. Dentro de estos filtros y transformaciones se encuentran, entre otros, el reescalado de la imagen, su binarización, el cambio de orientación, el aclarado u oscurecimiento de la imagen, su conversión a blanco y negro ...

Esta tarea se puede automatizar con el uso de librerías de Visión Computacional, como OpenCV.

- **Localización del texto:** este subproceso consiste en la detección de las áreas dentro de la imagen donde se encuentra el texto que se desea procesar. También puede ser llevado a cabo con la ayuda de una librería de Visión Computacional, utilizando técnicas de Deep Learning para reconocer donde aparecen caracteres dentro de la imagen.

El objetivo será acotar las áreas donde se encuentran dichos caracteres para derivar las regiones de interés (ROIs) de la imagen que nos interesan. Tras ello, dichos fragmentos serán pasados al siguiente subproceso del pipeline de OCR.

- **Segmentación de caracteres:** este subproceso parte de las regiones de interés derivadas en el subproceso anterior y consiste en la separación de las áreas correspondientes a cada uno de los caracteres localizados en el texto. Esta segmentación divide cada región de interés anterior en sus partes unitarias correspondientes a cada carácter.

En esta etapa también suele incluirse la segmentación del texto en líneas y palabras a través de la detección de los espacios verticales y horizontales o los signos de puntuación.

- **Reconocimiento de caracteres:** esta etapa trata el proceso de identificación de los caracteres concretos que aparecen representados en la imagen. Para ello, se hace uso de modelos de aprendizaje preentrenados con extensos datasets de caracteres. Los modelos utilizados para este fin estarán condicionados por características como el sistema de escritura del texto de la imagen. Por ejemplo, un modelo entrenado con símbolos del alfabeto cirílico no se podrá aplicar a un texto escrito en chino.

Este subproceso permite componer un texto preliminar a partir de la obtención de cada uno de los caracteres.

- **Postprocesamiento:** este último subproceso acomete la corrección del texto obtenido en el paso anterior. Consiste en la detección de los errores cometidos por el modelo anterior a la hora de predecir los caracteres del texto.

De una manera más concisa, el proceso OCR también se podría generalizar en 2 partes diferenciadas:

- **Detección del texto:** este primer paso agruparía las tareas para la detección de las regiones con texto de la imagen.
- **Reconocimiento del texto:** este segundo paso trataría el reconocimiento concreto de los caracteres contenidos en las regiones de interés de la imagen y la derivación del texto resultado del proceso.

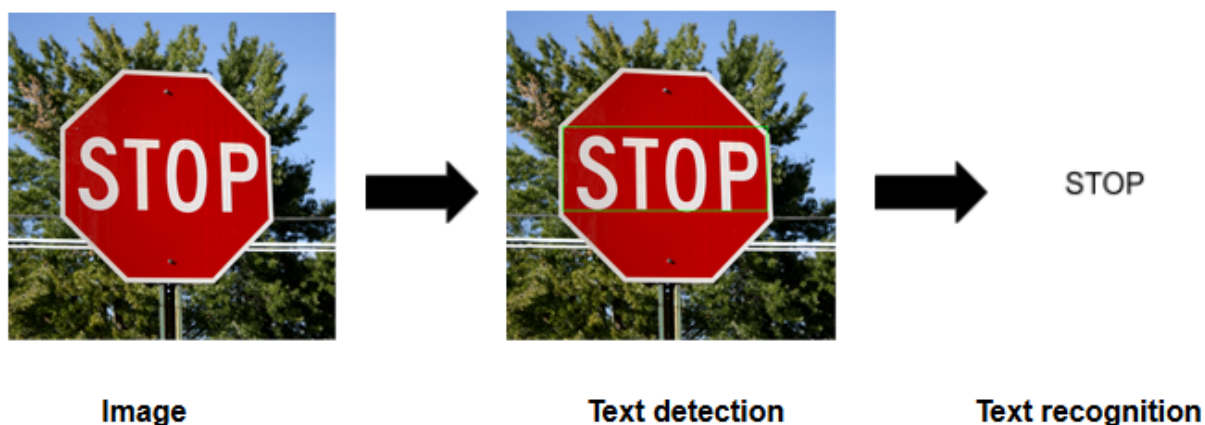


Figura 4.3: Proceso General OCR

Con vistas a ofrecer un resultado lo más acertado posible, el motor OCR contrasta el texto preliminar utilizando un modelo de Deep Learning entrenado con palabras y estructuras del lenguaje que evalúa la corrección de las partes del texto para determinar cuáles tienen más probabilidad de contener errores. A partir de ahí, el modelo podrá ajustar las partes más erróneas con otras propuestas más adecuadas para reducir el error cometido en el texto. Finalmente, la representación de texto refinada será la salida proporcionada por el motor OCR para la imagen pasada como entrada.

Como ya se ha mencionado, este proceso OCR seguido debe ajustarse a características concretas de la situación en la que se emplea. Por ejemplo, dependiendo del lenguaje en el que esté escrito el texto que se desea extraer, se deberá emplear un modelo de lenguaje concreto para mejorar la salida del proceso OCR.

Los paquetes de lenguaje aportados a los modelos proveen datasets con textos, palabras, símbolos o estructuras propias del lenguaje considerado que resultan de gran utilidad a las redes recurrentes LSTM para derivar de la forma más acertada posible la representación textual correcta. Esto se debe a que un idioma tiene implícitas estructuras propias que hacen posible predecir ciertas secuencias de caracteres que pueden seguir a otras aparecidas anteriormente. Esta característica es de especial relevancia cuando el motor OCR debe procesar ficheros de baja calidad o con erratas en los que sea difícil derivar los caracteres correctos sin considerar el contexto en el que suceden. Para ello, los motores OCR permiten instalar paquetes de lenguaje adicionales para procesar textos escritos en distintos idiomas, teniendo actualmente la mayoría de motores soporte para cientos de lenguajes.

Los usos actuales de esta tecnología son muy diversos. Entre otros muchos se encuentran los siguientes.

- Digitalización de documentos manuscritos
- Digitalización de cartas de restaurantes
- Reconocimiento de matrículas de vehículos
- Reconocimiento de señales de tráfico
- Extracción de cantidades en facturas
- Extracción de datos de pasaportes en aerolíneas
- Lectura de contadores eléctricos
- Automatización de la captura y procesado de datos en formularios

### 4.5. Aplicación Web

Una aplicación web es un tipo de software que se ejecuta en un servidor web y cuyo contenido está codificado de forma que pueda ser renderizado por un navegador web.

De un modo resumido, su funcionamiento se basa en el envío de peticiones y respuestas HTTP, según se muestra en la Figura 4.4.

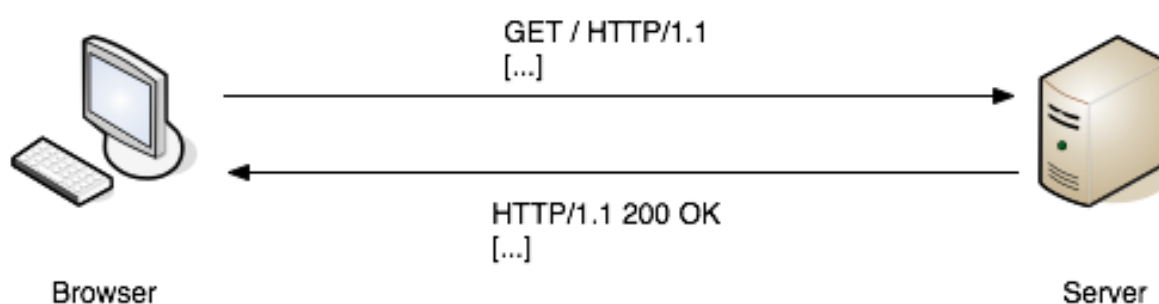


Figura 4.4: Flujo de Proceso HTTP

HTTP (Hypertext Transfer Protocol) es un protocolo de comunicación para la transferencia de información en la web. Las peticiones de recursos HTTP son enviadas desde un cliente (comúnmente un navegador) hasta el servidor y tienen una estructura definida compuesta por las siguientes partes:

- **Versión:** del protocolo HTTP utilizado para la petición.

- **Método:** según el propósito que se persiga con la petición.

Principalmente, se utilizan los 5 siguientes métodos de petición:

- **GET:** solicita la representación de un recurso en el servidor.
  - **HEAD:** igual que el método GET pero sin devolver el cuerpo de la respuesta. Comúnmente utilizado para monitorización.
  - **POST:** envía entidades con datos hacia una ruta de recurso en el servidor.
  - **PUT:** reemplaza la representación de un recurso en el servidor con la entidad de datos enviada en el cuerpo de la petición. En caso de no existir, se crearía.
  - **DELETE:** solicita la eliminación de un recurso en el servidor.
- **URL del Recurso:** la ruta en el servidor del recurso requerido.
  - **Cabeceras:** parámetros con información adicional acerca de la petición.
  - **Cuerpo:** contenido a enviar al servidor con la petición. Siendo opcional en caso de utilizarse los métodos GET, HEAD o DELETE.

Por otra parte, las respuestas HTTP son enviadas desde el servidor al cliente, correspondiéndose unívocamente cada una con su petición asociada. La estructura que tienen estas respuestas HTTP es la siguiente:

- **Versión:** del protocolo HTTP utilizado para la respuesta.
- **Código de estado:** indica si se ha completado satisfactoriamente la solicitud HTTP asociada.

Dentro de los códigos de estado existen varias categorías:

- **1xx:** para respuestas informativas.
  - **2xx:** para respuestas satisfactorias.
  - **3xx:** para redirecciones.
  - **4xx:** para errores en la petición enviada por el cliente.
  - **5xx:** para errores en el servidor.
- **Cabeceras:** parámetros con información adicional de la respuesta.
  - **Cuerpo:** contenido devuelto por el servidor asociado con el recurso solicitado.

Normalmente, de cara a la implementación de una aplicación web es recomendable considerar el uso de algún framework web que facilite su desarrollo. Un framework es un marco de trabajo que aglutina un conjunto de conceptos, criterios y buenas prácticas, como el uso de patrones de diseño, que apoyan el desarrollo y mantenimiento sostenible de un sistema. Gracias a estas características, los frameworks web se plantean como soluciones idóneas de cara a posibilitar un desarrollo rápido y adecuado de nuestra aplicación web.



# Capítulo 5

## Análisis

Annotator tiene como finalidad ser una herramienta de anotación de textos simple y rápida en la que se puedan gestionar diferentes proyectos de anotación a nivel de un único usuario.

Su desarrollo web abre la posibilidad para un futuro despliegue a gran escala en el que plantear la gestión de cuentas de distintos usuarios de la herramienta.

Acorde a la visión de esta primera versión del producto, se realiza un análisis teniendo presentes estos condicionantes comentados.

### 5.1. Actores del Sistema

Debido a la naturaleza monousuario de la herramienta, consideraremos un único actor en el sistema, al que nombraremos de forma genérica como Usuario.

ACT-01	Usuario
Descripción	Este actor representa el usuario fundamental de la herramienta, siendo el protagonista de todos los casos de uso involucrados en ella.
Notas	Todo el uso de la herramienta Annotator recae sobre este actor.

Tabla 5.1: Actor ACT-01 - Usuario

### 5.2. Requisitos de Usuario

Los requisitos de usuario representan las expectativas y funcionalidades que los usuarios esperan de su interacción con el sistema.

Al basarse en Scrum nuestra metodología de trabajo UVagile, estos requisitos de usuario los representamos como Historias de Usuario (User Stories).

La notación empleada para especificar Historias de Usuario tiene la siguiente estructura:

- **COMO:** el rol involucrado dentro del sistema. Ej: usuario.
- **QUIERO:** el objetivo a cumplir. Ej: añadir una nueva categoría de entidad.
- **PARA:** la motivación subyacente. Ej: anotar entidades con ella.

Por tanto, la Historia de Usuario ejemplificada quedaría así:

*COMO usuario QUIERO añadir una nueva categoría de entidad PARA anotar entidades con ella.*

A continuación, se especifican los requisitos de usuario identificados para la herramienta.

<b>ID Requisito</b>	<b>Descripción Requisito</b>
RU-01	COMO usuario QUIERO añadir una nueva anotación PARA incluirla dentro de las anotaciones realizadas.
RU-02	COMO usuario QUIERO eliminar una anotación realizada PARA no considerarla.
RU-03	COMO usuario QUIERO editar una anotación realizada PARA corregir sus límites en el texto.
RU-04	COMO usuario QUIERO cambiar la categoría de una anotación PARA corregirla.
RU-05	COMO usuario QUIERO añadir una nueva categoría de entidad PARA anotar entidades con ella.
RU-06	COMO usuario QUIERO eliminar una categoría existente PARA no considerarla más dentro del proceso de anotación.
RU-07	COMO usuario QUIERO seleccionar una de las categorías de entidad existentes PARA anotar las siguientes entidades con ella.
RU-08	COMO usuario QUIERO modificar el nombre de una categoría de entidad PARA cambiarlo a conveniencia.
RU-09	COMO usuario QUIERO establecer el color de una categoría de entidad PARA personalizar la visión de las entidades anotadas con ella.
RU-10	COMO usuario QUIERO crear una expresión regular PARA añadirla a las definidas.
RU-11	COMO usuario QUIERO eliminar una expresión regular PARA ya no considerarla.
RU-12	COMO usuario QUIERO editar una expresión regular PARA establecer el nombre y definición de expresión regular que necesite.
RU-13	COMO usuario QUIERO asociar una expresión regular a una categoría existente PARA determinar el patrón que debe seguir la categoría.
RU-14	COMO usuario QUIERO exportar las expresiones regulares PARA obtener las expresiones regulares definidas en el proyecto.

Tabla 5.2: Requisitos de Usuario



<b>ID Requisito</b>	<b>Descripción Requisito</b>
RU-15	COMO usuario QUIERO importar expresiones regulares PARA cargarlas en la interfaz de usuario y poder asociarlas a las categorías.
RU-16	COMO usuario QUIERO importar un nuevo documento de texto PARA incluirlo en el proceso de anotación.
RU-17	COMO usuario QUIERO importar un fichero PDF PARA extraer su texto e incluirlo en el proceso de anotación.
RU-18	COMO usuario QUIERO exportar los documentos anotados PARA obtener todas las anotaciones realizadas sobre el dataset.
RU-19	COMO usuario QUIERO navegar entre los documentos cargados PARA cambiar el texto sobre el que anotar.
RU-20	COMO usuario QUIERO navegar entre las sentencias de un documento PARA cambiar la sentencia sobre la que anotar.
RU-21	COMO usuario QUIERO importar un modelo PARA aplicar la anotación asistida con él.
RU-22	COMO usuario QUIERO exportar el modelo de aprendizaje PARA obtener el modelo entrenado durante el proceso de anotación.
RU-23	COMO usuario QUIERO activar la asistencia por patrones de coincidencia PARA encontrar las entidades que coinciden con mis expresiones regulares.
RU-24	COMO usuario QUIERO activar la asistencia mediante el entrenamiento de un modelo con las anotaciones realizadas PARA predecir las entidades nombradas de un texto con mayor eficacia.
RU-25	COMO usuario QUIERO activar la asistencia mediante un modelo importado PARA obtener entidades a partir de las predicciones del modelo.
RU-26	COMO usuario QUIERO visualizar estadísticas de anotación PARA conocer el estado actual del proceso de anotación.
RU-27	COMO usuario QUIERO visualizar resultados de entrenamiento PARA conocer datos del entrenamiento de modelo realizado.
RU-28	COMO usuario QUIERO establecer opciones de importación PARA personalizar el proceso de importación de documentos.

Tabla 5.3: Requisitos de Usuario

### 5.3. Casos de Uso

Los casos de uso representan las acciones o actividades que pueden realizar los usuarios en su manipulación del sistema.

Para nuestra herramienta, los casos de uso que han sido identificados son los siguientes:

<b>ID Caso de Uso</b>	<b>Nombre Caso de Uso</b>
CU-01	Crear un proyecto de anotación.
CU-02	Abrir un proyecto de anotación.
CU-03	Renombrar un proyecto de anotación.
CU-04	Eliminar un proyecto de anotación.
CU-05	Guardar un proyecto de anotación.
CU-06	Añadir una anotación.
CU-07	Eliminar una anotación.
CU-08	Editar los límites de una anotación.
CU-09	Cambiar la categoría de una anotación.
CU-10	Añadir una categoría de entidad.
CU-11	Eliminar una categoría de entidad.
CU-12	Editar el nombre de una categoría de entidad.
CU-13	Editar el color de una categoría de entidad.
CU-14	Seleccionar una categoría de entidad.
CU-15	Crear una expresión regular.
CU-16	Eliminar una expresión regular.
CU-17	Editar una expresión regular.
CU-18	Asociar una expresión regular a una categoría de entidad.
CU-19	Exportar fichero con expresiones regulares.
CU-20	Importar fichero con expresiones regulares.

Tabla 5.4: Casos de Uso

<b>ID Caso de Uso</b>	<b>Nombre Caso de Uso</b>
CU-21	Importar documento de texto.
CU-22	Importar fichero PDF.
CU-23	Importar documentos anotados.
CU-24	Exportar documentos anotados.
CU-25	Navegar entre los documentos.
CU-26	Navegar entre las sentencias de un documento.
CU-27	Importar modelo de aprendizaje preentrenado.
CU-28	Exportar modelo de aprendizaje entrenado.
CU-29	Activar asistencia con patrones de coincidencia.
CU-30	Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia.
CU-31	Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia.
CU-32	Visualizar estadísticas de anotación del proyecto.
CU-33	Visualizar resultados del entrenamiento de un modelo de aprendizaje.
CU-34	Establecer opciones de importación.

Tabla 5.5: Casos de Uso

Igualmente, en la Figura 5.1 se presenta el diagrama de casos de uso completo asociado a la herramienta. Algunos de los casos de uso han sido agrupados en categorías comunes con el fin de obtener un diagrama más estructurado.

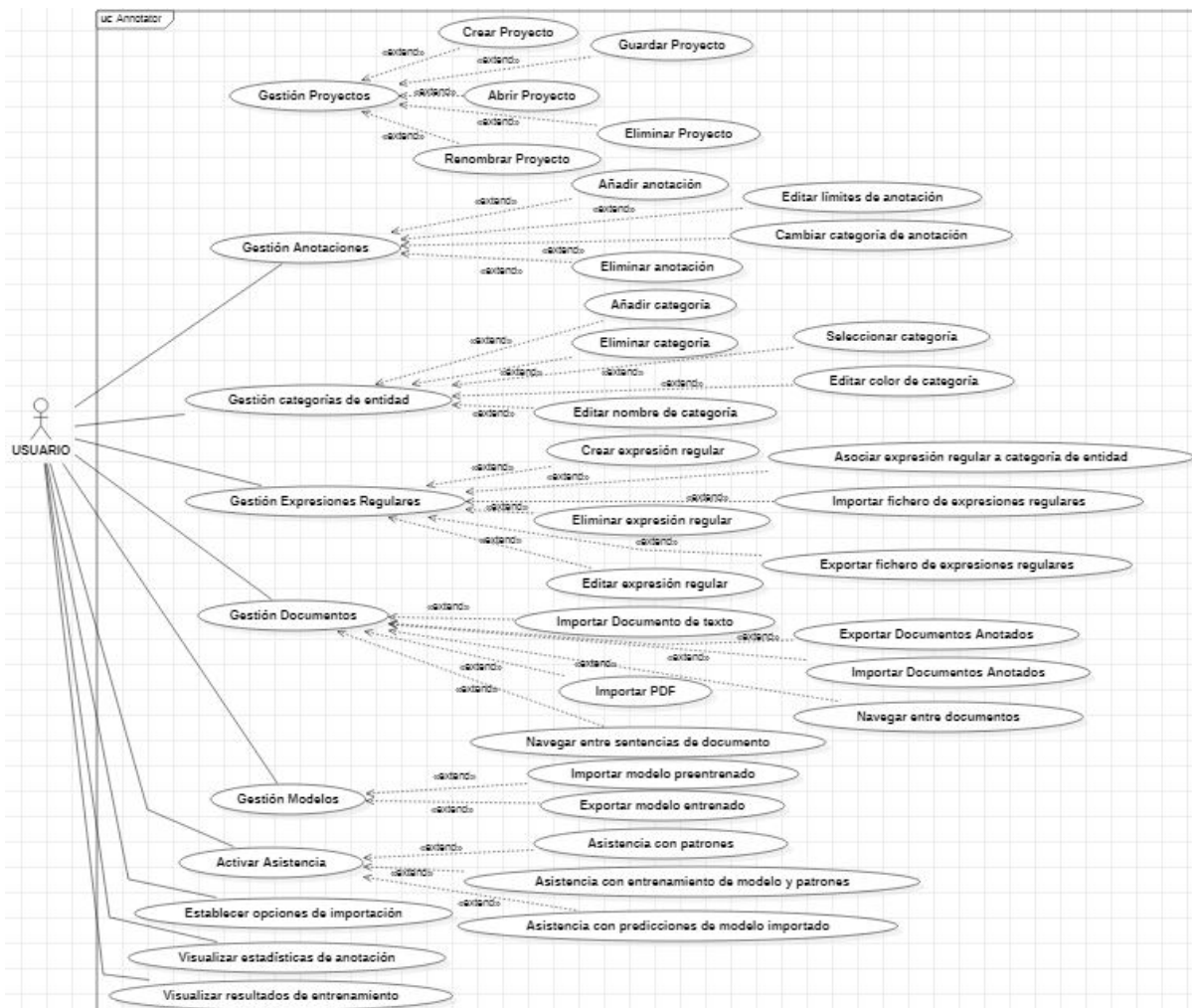


Figura 5.1: Diagrama de Casos de Uso

### 5.3.1. Especificación de Casos de Uso

En este apartado se presentará la especificación de algunos de los casos de uso identificados para nuestra herramienta.

La especificación del resto de casos de uso contemplados se obvia por analogía con los expuestos a continuación.

<b>CU-01</b>	<b>Crear un proyecto de anotación</b>
Descripción	El usuario crea un nuevo proyecto de anotación en la herramienta.
Precondición	Ninguna.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario solicita crear un nuevo proyecto.</li> <li>2. El sistema pide al usuario la introducción de un nombre para el proyecto.</li> <li>3. El usuario introduce el nombre para el nuevo proyecto.</li> <li>4. El sistema verifica la corrección del nombre introducido por el usuario.</li> <li>5. El sistema crea el nuevo proyecto de anotación.</li> <li>6. El sistema redirige al usuario a la página principal de la herramienta.</li> </ol>
Postcondición	Proyecto de anotación creado satisfactoriamente.
Excepciones	<ol style="list-style-type: none"> <li>4a) El usuario introduce un nombre de proyecto ya existente o un nombre inválido.</li> <li>3b) El sistema muestra un mensaje de advertencia comunicando el error.</li> </ol>
Prioridad	Media

Tabla 5.6: Especificación Caso de Uso CU-01: Crear un proyecto de anotación.

<b>CU-02</b>	<b>Abrir un proyecto de anotación</b>
Descripción	El usuario abre un proyecto de anotación existente en la herramienta.
Precondición	Ninguna.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario solicita abrir un nuevo proyecto.</li> <li>2. El sistema muestra un listado con los nombres de proyectos existentes en la herramienta.</li> <li>3. El usuario selecciona uno de los proyectos del listado.</li> <li>4. El sistema carga el proyecto seleccionado en la herramienta.</li> <li>5. El sistema redirige al usuario a la página principal de la herramienta.</li> </ol>
Postcondición	Proyecto de anotación cargado en la herramienta satisfactoriamente.
Excepciones	3a) El sistema muestra un listado vacío al no existir ningún proyecto de anotación en la herramienta.
Prioridad	Media

Tabla 5.7: Especificación Caso de Uso CU-02: Abrir un proyecto de anotación.

<b>CU-03</b>	<b>Renombrar un proyecto de anotación</b>
Descripción	El usuario renombra un proyecto de anotación.
Precondición	El usuario ha abierto un proyecto de anotación.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario solicita renombrar el proyecto actual.</li> <li>2. El sistema pide la introducción del nuevo nombre al usuario.</li> <li>3. El usuario introduce el nuevo nombre del proyecto actual.</li> <li>4. El sistema verifica la corrección del nuevo nombre introducido por el usuario.</li> <li>5. El sistema cambia el nombre del proyecto actual.</li> </ol>
Postcondición	Proyecto de anotación renombrado satisfactoriamente.
Excepciones	<ol style="list-style-type: none"> <li>4a) El nombre de proyecto introducido por el usuario corresponde a otro proyecto existente o es un nombre inválido.</li> <li>4b) El sistema muestra un mensaje de advertencia comunicando el error.</li> </ol>
Prioridad	Media

Tabla 5.8: Especificación Caso de Uso CU-03: Renombrar un proyecto de anotación.

<b>CU-04</b>	<b>Eliminar un proyecto de anotación</b>
Descripción	El usuario elimina un proyecto de anotación.
Precondición	El usuario ha abierto un proyecto de anotación.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario solicita eliminar el proyecto actual.</li> <li>2. El sistema pide confirmación al usuario antes de eliminar el proyecto actual.</li> <li>3. El usuario confirma la decisión de eliminar el proyecto actual.</li> <li>4. El sistema elimina de la herramienta el proyecto actual.</li> </ol>
Postcondición	Proyecto de anotación eliminado satisfactoriamente de la herramienta.
Excepciones	3a) El usuario finalmente decide no eliminar el proyecto actual.
Prioridad	Media

Tabla 5.9: Especificación Caso de Uso CU-04: Eliminar un proyecto de anotación.

<b>CU-06</b>	<b>Añadir una anotación</b>
Descripción	El usuario añade una nueva anotación de entidad en un texto.
Precondición	<p>El usuario ha abierto un proyecto de anotación.</p> <p>El usuario ha importado algún documento al proyecto.</p>
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una categoría de entidad con la que realizar la anotación.</li> <li>2. El usuario selecciona con el ratón una entidad en el texto.</li> <li>3. El sistema verifica la corrección de la anotación realizada sobre el texto por el usuario.</li> <li>4. El sistema crea la anotación seleccionada por el usuario y la asigna la categoría de entidad correspondiente.</li> </ol>
Postcondición	Anotación añadida satisfactoriamente.
Excepciones	<ol style="list-style-type: none"> <li>1a) El usuario no puede seleccionar una categoría de entidad al no existir ninguna.</li> <li>1b) El usuario procede a crear una nueva categoría de entidad.</li> <li>3a) La anotación realizada no es válida o colisiona con otra anotación existente.</li> <li>3b) El sistema muestra un mensaje de advertencia comunicando el error.</li> </ol>
Prioridad	Muy Alta

Tabla 5.10: Especificación Caso de Uso CU-06: Añadir una anotación.

<b>CU-08</b>	<b>Editar los límites de una anotación</b>
Descripción	El usuario edita los límites de una anotación existente.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha importado algún documento al proyecto. El usuario ha realizado alguna anotación previamente.
Secuencia Normal	1. El usuario modifica los límites de una anotación existente. 2. El sistema verifica la corrección de la anotación con los nuevos límites. 3. El sistema cambia los límites de la anotación en cuestión.
Postcondición	Anotación modificada satisfactoriamente.
Excepciones	2a) El sistema comprueba que los nuevos límites de la anotación no son válidos o colisiona con los de otra anotación existente. 2b) El sistema muestra un mensaje de advertencia comunicando el error.
Prioridad	Media

Tabla 5.11: Especificación Caso de Uso CU-08: Editar los límites de una anotación.

<b>CU-11</b>	<b>Eliminar una categoría de entidad</b>
Descripción	El usuario elimina una categoría de entidad existente.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha creado alguna categoría de entidad previamente.
Secuencia Normal	1. El usuario selecciona la eliminación de una categoría de entidad. 2. El sistema muestra al usuario un mensaje de confirmación de eliminación en caso de que la categoría de entidad tuviera alguna anotación asociada. 3. El usuario decide eliminar la categoría de entidad. 4. El sistema elimina la categoría de entidad del proyecto actual.
Postcondición	Categoría de entidad eliminada del proyecto.
Excepciones	3a) El usuario decide no eliminar la categoría de entidad.
Prioridad	Media

Tabla 5.12: Especificación Caso de Uso CU-11: Eliminar una categoría de entidad.



<b>CU-13</b>	<b>Editar el color de una categoría de entidad</b>
Descripción	El usuario cambia el color asociado a una categoría de entidad existente.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha creado alguna categoría de entidad previamente.
Secuencia Normal	1. El usuario selecciona un nuevo color para la categoría de entidad. 2. El sistema verifica la corrección del cambio de color de la categoría de entidad. 3. El sistema cambia el color de la categoría de entidad y de todas sus anotaciones asociadas.
Postcondición	Color de la categoría de entidad y de sus anotaciones asociadas modificado.
Excepciones	2a) El sistema comprueba que el nuevo color ya está asignado a otra categoría de entidad. 2b) El sistema muestra un mensaje de advertencia comunicando el error.
Prioridad	Media

Tabla 5.13: Especificación Caso de Uso CU-13: Editar el color de una categoría de entidad.

<b>CU-21</b>	<b>Importar documento de texto</b>
Descripción	El usuario importa al proyecto actual un documento de texto para anotar.
Precondición	El usuario ha abierto un proyecto de anotación.
Secuencia Normal	1. El usuario selecciona importar un nuevo documento de texto. 2. El sistema muestra un espacio para cargar el documento de texto. 3. El usuario sube el documento de texto a la herramienta. 4. El sistema valida la corrección del fichero subido por el usuario a la herramienta. 5. El sistema carga el texto en el proyecto actual para anotarlo.
Postcondición	Texto cargado en el proyecto actual para su anotación.
Excepciones	4a) El sistema comprueba que el fichero subido por el usuario a la herramienta no tiene una extensión válida. 4b) El sistema muestra un mensaje de advertencia comunicando el error.
Prioridad	Muy Alta

Tabla 5.14: Especificación Caso de Uso CU-21: Importar documento de texto.

<b>CU-22</b>	<b>Importar fichero PDF</b>
Descripción	El usuario importa al proyecto actual un fichero PDF.
Precondición	El usuario ha abierto un proyecto de anotación.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona importar un fichero PDF.</li> <li>2. El sistema muestra un espacio para cargar el fichero PDF.</li> <li>3. El usuario sube el fichero PDF a la herramienta.</li> <li>4. El sistema valida la corrección del fichero subido por el usuario a la herramienta.</li> <li>5. El sistema procesa el fichero PDF mediante la tecnología OCR para obtener el texto contenido.</li> <li>6. El sistema carga el texto en el proyecto actual para anotarlo.</li> </ol>
Postcondición	Texto cargado en el proyecto actual para su anotación.
Excepciones	<ol style="list-style-type: none"> <li>4a) El sistema comprueba que el fichero subido por el usuario a la herramienta no tiene la extensión .pdf.</li> <li>4b) El sistema muestra un mensaje de advertencia comunicando el error.</li> </ol>
Prioridad	Alta

Tabla 5.15: Especificación Caso de Uso CU-22: Importar fichero PDF.

<b>CU-23</b>	<b>Importar documentos anotados</b>
Descripción	El usuario importa documentos anotados al proyecto actual.
Precondición	El usuario ha abierto un proyecto de anotación.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona importar un dataset anotado al proyecto actual.</li> <li>2. El sistema muestra un espacio para cargar el dataset anotado.</li> <li>3. El usuario sube el dataset anotado a la herramienta.</li> <li>4. El sistema valida la corrección del fichero subido por el usuario a la herramienta.</li> <li>5. El sistema procesa el dataset anotado según el tipo de fichero subido por el usuario y lo carga en el proyecto actual para su anotación.</li> </ol>
Postcondición	Textos anotados cargados en el proyecto actual.
Excepciones	<ol style="list-style-type: none"> <li>4a) El sistema comprueba que el fichero subido por el usuario a la herramienta no tiene una extensión válida.</li> <li>4b) El sistema muestra un mensaje de advertencia comunicando el error.</li> <li>5a) El sistema no puede procesar el fichero subido por el usuario al no ser éste válido.</li> <li>5b) El sistema muestra un mensaje de error al no poder llevar a cabo la operación.</li> </ol>
Prioridad	Alta

Tabla 5.16: Especificación Caso de Uso CU-23: Importar documentos anotados.

<b>CU-24</b>	<b>Exportar documentos anotados</b>
Descripción	El usuario exporta los documentos anotados en el proyecto actual.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha importado algún documento al proyecto actual previamente.
Secuencia Normal	1. El usuario selecciona exportar el dataset anotado del proyecto actual. 2. El sistema consulta al usuario el formato en el que exportar el dataset. 3. El usuario decide el formato en el que exportar el dataset entre los disponibles. 4. El sistema genera el dataset anotado y lo descarga en el cliente.
Postcondición	Dataset exportado por la herramienta satisfactoriamente.
Excepciones	Ninguna.
Prioridad	Muy Alta

Tabla 5.17: Especificación Caso de Uso CU-24: Exportar documentos anotados.

<b>CU-29</b>	<b>Activar asistencia con patrones de coincidencia</b>
Descripción	El usuario activa el modo de asistencia a la anotación con patrones de coincidencia.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha importado algún documento al proyecto actual previamente.
Secuencia Normal	1. El usuario activa la asistencia a la anotación con patrones de coincidencia. 2. El sistema verifica que existe alguna expresión regular asociada a alguna categoría de entidad en el proyecto actual. 3. El sistema procesa los documentos del proyecto actual aplicando los patrones de coincidencia. 4. El sistema añade nuevas anotaciones en los documentos acorde a los patrones de coincidencia aplicados.
Postcondición	Nuevas anotaciones sugeridas acorde a los patrones de coincidencia aplicados.
Excepciones	2a) El sistema comprueba que no existe ninguna asociación entre expresión regular y categoría de entidad en el proyecto actual. 2b) El sistema muestra un mensaje de advertencia comunicando el error.
Prioridad	Alta

Tabla 5.18: Especificación Caso de Uso CU-29: Activar asistencia con patrones de coincidencia.

<b>CU-30</b>	<b>Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia</b>
Descripción	El usuario activa el modo de asistencia a la anotación con entrenamiento de modelo de aprendizaje y patrones de coincidencia.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha importado algún documento al proyecto actual previamente.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario activa la asistencia a la anotación con entrenamiento de modelo de aprendizaje y patrones de coincidencia.</li> <li>2. El sistema verifica que existe alguna anotación realizada sobre los documentos del proyecto actual.</li> <li>3. El sistema procesa los documentos del proyecto actual entrenando un modelo de aprendizaje con las anotaciones realizadas y aplicando los patrones de coincidencia.</li> <li>4. El sistema añade nuevas anotaciones en los documentos acorde al entrenamiento realizado y a los patrones de coincidencia aplicados.</li> <li>5. El sistema muestra resultados del entrenamiento del modelo llevado a cabo.</li> </ol>
Postcondición	Nuevas anotaciones sugeridas acorde al entrenamiento de modelo de aprendizaje realizado y a los patrones de coincidencia aplicados.
Excepciones	<ol style="list-style-type: none"> <li>2a) El sistema comprueba que no existe ninguna anotación realizada sobre los documentos del proyecto actual.</li> <li>2b) El sistema muestra un mensaje de advertencia comunicando el error.</li> </ol>
Prioridad	Alta

Tabla 5.19: Especificación Caso de Uso CU-30: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia.

<b>CU-31</b>	<b>Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia</b>
Descripción	El usuario activa el modo de asistencia a la anotación con predicciones de modelo de aprendizaje importado y patrones de coincidencia.
Precondición	El usuario ha abierto un proyecto de anotación. El usuario ha importado algún documento al proyecto actual previamente.
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El usuario activa la asistencia a la anotación con predicciones de modelo de aprendizaje importado y patrones de coincidencia.</li> <li>2. El sistema verifica que existe un modelo de aprendizaje importado en el proyecto actual.</li> <li>3. El sistema carga el modelo de aprendizaje importado y procesa los documentos del proyecto actual aplicando las predicciones del modelo de aprendizaje importado.</li> <li>4. El sistema añade nuevas anotaciones en los documentos acorde a las predicciones del modelo de aprendizaje importado.</li> </ol>
Postcondición	Nuevas anotaciones sugeridas acorde a las predicciones del modelo de aprendizaje importado.
Excepciones	<ol style="list-style-type: none"> <li>2a) El sistema comprueba que no existe ningún modelo de aprendizaje importado en el proyecto actual.</li> <li>2b) El sistema muestra un mensaje de advertencia comunicando el error.</li> </ol>
Prioridad	Alta

Tabla 5.20: Especificación Caso de Uso CU-31: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia.

## 5.4. Requisitos Funcionales

Los requisitos funcionales describen las funcionalidades que ofrece el sistema en su interacción con el usuario.

Estos requisitos representan las actividades y comportamiento básicos que debe cumplir el sistema y por las que se valora principalmente su utilidad.

A continuación, se obtienen los requisitos funcionales del sistema derivados de cada uno de los casos de uso identificados para la herramienta.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-01	El sistema mostrará la opción para crear un proyecto de anotación.
RF-02	El sistema habilitará un campo de texto para introducir el nombre de un proyecto.
RF-03	El sistema creará el proyecto de anotación con el nombre establecido por el usuario.

Tabla 5.21: Requisitos Funcionales relativos al Caso de Uso CU-01: Crear un proyecto de anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-04	El sistema mostrará la opción para abrir un proyecto de anotación.
RF-05	El sistema habilitará un campo de selección para elegir el proyecto de anotación que abrir.
RF-06	El sistema abrirá el proyecto de anotación en la herramienta.

Tabla 5.22: Requisitos Funcionales relativos al Caso de Uso CU-02: Abrir un proyecto de anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-07	El sistema mostrará la opción para renombrar el proyecto de anotación actual.
RF-08	El sistema habilitará un campo de texto para introducir el nuevo nombre del proyecto de anotación actual.
RF-09	El sistema renombrará el proyecto de anotación actual.

Tabla 5.23: Requisitos Funcionales relativos al Caso de Uso CU-03: Renombrar un proyecto de anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-10	El sistema mostrará la opción para eliminar el proyecto de anotación actual.
RF-11	El sistema eliminará el proyecto de anotación actual.

Tabla 5.24: Requisitos Funcionales relativos al Caso de Uso CU-04: Eliminar un proyecto de anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-12	El sistema mostrará la opción para guardar el proyecto de anotación actual.
RF-13	El sistema guardará la información del proyecto de anotación actual.

Tabla 5.25: Requisitos Funcionales relativos al Caso de Uso CU-05: Guardar un proyecto de anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-14	El sistema comprobará la validez de la anotación realizada por el usuario.
RF-15	El sistema creará la anotación asociada a la categoría de entidad seleccionada.
RF-16	El sistema mostrará un mensaje de advertencia comunicando el error.

Tabla 5.26: Requisitos Funcionales relativos al Caso de Uso CU-06: Añadir una anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-17	El sistema mostrará la opción para eliminar una anotación del proyecto.
RF-18	El sistema eliminará la anotación seleccionada por el usuario.

Tabla 5.27: Requisitos Funcionales relativos al Caso de Uso CU-07: Eliminar una anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-19	El sistema comprobará la validez de los nuevos límites de la anotación modificados por el usuario.
RF-20	El sistema modificará los límites asociados a la anotación.
RF-16	El sistema mostrará un mensaje de advertencia comunicando el error.

Tabla 5.28: Requisitos Funcionales relativos al Caso de Uso CU-08: Editar los límites de una anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-21	El sistema cambiará la categoría de entidad asociada a la anotación.

Tabla 5.29: Requisitos Funcionales relativos al Caso de Uso CU-09: Cambiar la categoría de una anotación.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-22	El sistema mostrará la opción para añadir una nueva categoría de entidad.
RF-23	El sistema comprobará la validez del nombre introducido por el usuario para la categoría de entidad.
RF-24	El sistema comprobará la validez del color introducido por el usuario para la categoría de entidad.
RF-25	El sistema creará la nueva categoría de entidad.
RF-16	El sistema mostrará un mensaje de advertencia comunicando el error.

Tabla 5.30: Requisitos Funcionales relativos al Caso de Uso CU-10: Añadir una categoría de entidad.



<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-26	El sistema mostrará la opción para eliminar una categoría de entidad.
RF-27	El sistema solicitará al usuario confirmación de la eliminación de la categoría de entidad, en caso de tener alguna anotación asociada.
RF-28	El sistema eliminará la categoría de anotación.

Tabla 5.31: Requisitos Funcionales relativos al Caso de Uso CU-11: Eliminar una categoría de entidad.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-29	El sistema comprobará la validez del nuevo nombre introducido por el usuario para la categoría de entidad.
RF-30	El sistema cambiará el nombre de la categoría de entidad.

Tabla 5.32: Requisitos Funcionales relativos al Caso de Uso CU-12: Editar el nombre de una categoría de entidad.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-31	El sistema comprobará la validez del nuevo color introducido por el usuario para la categoría de entidad.
RF-32	El sistema cambiará el color de la categoría de entidad.

Tabla 5.33: Requisitos Funcionales relativos al Caso de Uso CU-13: Editar el color de una categoría de entidad.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-33	El sistema seleccionará la categoría de entidad escogida por el usuario para la anotación de próximas anotaciones.
RF-34	El sistema cambiará el color de la categoría de entidad.

Tabla 5.34: Requisitos Funcionales relativos al Caso de Uso CU-14: Seleccionar una categoría de entidad.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-35	El sistema mostrará la opción para crear una nueva expresión regular.
RF-36	El sistema comprobará la validez de la nueva expresión regular introducida por el usuario.
RF-37	El sistema creará la expresión regular.

Tabla 5.35: Requisitos Funcionales relativos al Caso de Uso CU-15: Crear una expresión regular.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-38	El sistema mostrará la opción para eliminar una nueva expresión regular.
RF-39	El sistema eliminará la expresión regular.

Tabla 5.36: Requisitos Funcionales relativos al Caso de Uso CU-16: Eliminar una expresión regular.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-40	El sistema comprobará la validez del nombre y definición nuevos modificados por el usuario.
RF-41	El sistema modificará la expresión regular.

Tabla 5.37: Requisitos Funcionales relativos al Caso de Uso CU-17: Editar una expresión regular.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-42	El sistema asociará la expresión regular a la categoría de entidad.

Tabla 5.38: Requisitos Funcionales relativos al Caso de Uso CU-18: Asociar una expresión regular a una categoría de entidad.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-43	El sistema mostrará la opción para exportar un fichero de expresiones regulares.
RF-44	El sistema exportará al cliente un fichero de expresiones regulares.

Tabla 5.39: Requisitos Funcionales relativos al Caso de Uso CU-19: Exportar fichero con expresiones regulares.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-45	El sistema mostrará la opción para importar un fichero de expresiones regulares.
RF-46	El sistema comprobará la validez del fichero cargado por el usuario.
RF-47	El sistema importará el fichero de expresiones regulares cargado por el usuario.

Tabla 5.40: Requisitos Funcionales relativos al Caso de Uso CU-20: Importar fichero con expresiones regulares.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-48	El sistema mostrará la opción para importar un documento de texto.
RF-46	El sistema comprobará la validez del fichero cargado por el usuario.
RF-49	El sistema cargará en la interfaz el texto resultante del fichero.

Tabla 5.41: Requisitos Funcionales relativos al Caso de Uso CU-21: Importar documento de texto.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-50	El sistema mostrará la opción para importar un fichero PDF.
RF-46	El sistema comprobará la validez del fichero cargado por el usuario.
RF-51	El sistema procesará mediante OCR el fichero PDF para obtener el texto contenido.
RF-49	El sistema cargará en la interfaz el texto resultante del fichero.

Tabla 5.42: Requisitos Funcionales relativos al Caso de Uso CU-22: Importar fichero PDF.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-54	El sistema mostrará la opción para importar documentos anotados.
RF-46	El sistema comprobará la validez del fichero cargado por el usuario.
RF-55	El sistema cargará en la interfaz los documentos anotados resultantes del fichero.

Tabla 5.43: Requisitos Funcionales relativos al Caso de Uso CU-23: Importar documentos anotados.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-56	El sistema mostrará la opción para exportar los documentos del proyecto.
RF-52	El sistema solicitará al usuario elegir el formato en el que exportar los documentos del proyecto.
RF-57	El sistema exportará al cliente un fichero con los documentos del proyecto en el formato elegido por el usuario.

Tabla 5.44: Requisitos Funcionales relativos al Caso de Uso CU-24: Exportar documentos anotados.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-58	El sistema mostrará opciones para navegar entre los documentos del proyecto.
RF-59	El sistema cargará en la vista central la primera sentencia del documento seleccionado.

Tabla 5.45: Requisitos Funcionales relativos al Caso de Uso CU-25: Navegar entre los documentos.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-60	El sistema mostrará opciones para navegar entre las sentencias de un documento del proyecto.
RF-61	El sistema cargará en la vista central la sentencia seleccionada del documento.

Tabla 5.46: Requisitos Funcionales relativos al Caso de Uso CU-26: Navegar entre las sentencias de un documento.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-62	El sistema mostrará la opción para importar un modelo de aprendizaje.
RF-46	El sistema comprobará la validez del fichero cargado por el usuario.
RF-63	El sistema cargará en el proyecto actual el modelo de aprendizaje importado.

Tabla 5.47: Requisitos Funcionales relativos al Caso de Uso CU-27: Importar modelo de aprendizaje preentrenado.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-64	El sistema mostrará la opción para exportar el modelo de aprendizaje entrenado en el proyecto actual.
RF-65	El sistema exportará al cliente un fichero con el modelo de aprendizaje entrenado en el proyecto actual.

Tabla 5.48: Requisitos Funcionales relativos al Caso de Uso CU-28: Exportar modelo de aprendizaje entrenado.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-66	El sistema mostrará la opción para activar la asistencia con patrones de coincidencia.
RF-67	El sistema procesará los documentos del proyecto para la asistencia a la anotación.
RF-68	El sistema mostrará en la interfaz las anotaciones derivadas del proceso de asistencia a la anotación.

Tabla 5.49: Requisitos Funcionales relativos al Caso de Uso CU-29: Activar asistencia con patrones de coincidencia.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-69	El sistema mostrará la opción para activar la asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia.
RF-70	El sistema permitirá personalizar al usuario parámetros asociados al entrenamiento.
RF-71	El sistema entrenará un modelo de aprendizaje automático.
RF-67	El sistema procesará los documentos del proyecto para la asistencia a la anotación.
RF-68	El sistema mostrará en la interfaz las anotaciones derivadas del proceso de asistencia a la anotación.

Tabla 5.50: Requisitos Funcionales relativos al Caso de Uso CU-30: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-72	El sistema mostrará la opción para activar la asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia.
RF-67	El sistema procesará los documentos del proyecto para la asistencia a la anotación.
RF-68	El sistema mostrará en la interfaz las anotaciones derivadas del proceso de asistencia a la anotación.

Tabla 5.51: Requisitos Funcionales relativos al Caso de Uso CU-31: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-73	El sistema mostrará la opción para visualizar estadísticas de anotación del proyecto.
RF-74	El sistema generará una vista con las estadísticas de anotación del proyecto.

Tabla 5.52: Requisitos Funcionales relativos al Caso de Uso CU-32: Visualizar estadísticas de anotación del proyecto.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-75	El sistema mostrará la opción para visualizar resultados del entrenamiento del modelo de aprendizaje.
RF-76	El sistema generará una vista con los resultados del entrenamiento del modelo de aprendizaje entrenado en el proyecto.

Tabla 5.53: Requisitos Funcionales relativos al Caso de Uso CU-33: Visualizar resultados del entrenamiento de un modelo de aprendizaje.

<b>ID Requisito Funcional</b>	<b>Descripción Requisito Funcional</b>
RF-77	El sistema mostrará la opción para establecer las opciones personalizadas de importación de documentos.
RF-78	El sistema guardará las opciones personalizadas de importación en la configuración del proyecto actual.

Tabla 5.54: Requisitos Funcionales relativos al Caso de Uso CU-34: Establecer opciones de importación.



## 5.5. Requisitos No Funcionales

Los requisitos no funcionales engloban las propiedades que sirven para evaluar la operación con el sistema, tales como la usabilidad, el desempeño o la seguridad.

Estos requisitos, sin llegar a representar las funcionalidades primarias del sistema, son cruciales en el éxito del mismo e imprescindibles desde el punto de vista de la experiencia del usuario.

A continuación, se presentan los requisitos no funcionales identificados para nuestra herramienta.

<b>ID Requisito No Funcional</b>	<b>Descripción Requisito No Funcional</b>
RNF-01	La herramienta dispone de facilidades de ratón para la gestión de anotaciones de entidades.
RNF-02	Los textos importados a la herramienta presentan un interlineado pensado para facilitar su anotación.
RNF-03	La herramienta incluye metáforas visuales para mejorar la usabilidad y conseguir una interfaz más intuitiva.
RNF-04	La herramienta muestra mensajes informativos y de advertencia para mejorar su expresividad.
RNF-05	La herramienta presenta diferentes fuentes y tamaños de letra para mejorar la visibilidad de algunas de sus partes.
RNF-06	La herramienta hace uso de colores para mejorar el aspecto y usabilidad de la interfaz.
RNF-07	La herramienta permite mostrar/ocultar menús para permitir personalizar la organización de los elementos en la interfaz.

Tabla 5.55: Requisitos No Funcionales

## 5.6. Reglas de Negocio

Las reglas de negocio describen las políticas, estándares y restricciones organizacionales que rigen la definición e implementación del producto a desarrollar.

Para este proyecto, se han tenido en cuenta las siguientes reglas de negocio:

ID Regla de Negocio	Descripción Regla de Negocio
RN-01	La asignación del color inicial de una nueva categoría de entidad será automática a partir de un conjunto excluyente de colores preestablecidos.
RN-02	El conjunto de caracteres utilizables para formar parte del nombre de los distintos componentes de la herramienta está restringido.
RN-03	2 componentes dentro de un mismo proyecto de anotación no pueden tener el mismo nombre.
RN-04	2 categorías de entidad dentro de un mismo proyecto de anotación no pueden tener asociado el mismo color.
RN-05	La exportación de un dataset anotado seguirá el formato binario de entrenamiento de Spacy o el formato textual IOB2.
RN-06	La exportación de expresiones regulares seguirá un formato JSON propietario de la herramienta.
RN-07	La superposición de 2 anotaciones de entidades en un texto no está permitida.
RN-08	Una expresión regular debe estar asociada a una única categoría de entidad.
RN-09	La importación de nuevos documentos a un proyecto quedará bloqueada durante el proceso de asistencia a la anotación.
RN-10	La eliminación de una categoría de entidad con anotaciones asociadas requerirá de una confirmación por parte del usuario.
RN-11	La asistencia a la anotación sólo podrá ser activada si hay algún documento importado en el proyecto.
RN-12	El sistema deberá mostrar mensajes de seguimiento cuando se ejecuten procesos de larga duración.
RN-13	El idioma principal empleado en la herramienta será el Español.

Tabla 5.56: Reglas de Negocio

## 5.7. Requisitos de Información

Los requisitos de información representan los datos del dominio de la aplicación que el sistema debe almacenar para su uso efectivo.

De cara a identificar los componentes principales que conforman el dominio de información de la aplicación es de gran utilidad generar un modelo conceptual del sistema que identifique los conceptos básicos involucrados en su funcionalidad.

Una aproximación utilizada a tal efecto es el Diagrama Entidad-Relación.

El Diagrama Entidad-Relación es un diagrama conceptual que describe las entidades que definen los diferentes objetos de interés del sistema junto con las relaciones que deben establecerse entre ellos para satisfacer la funcionalidad requerida para el sistema.

A continuación, se plasma el Diagrama Entidad-Relación construido para la herramienta Annotator.

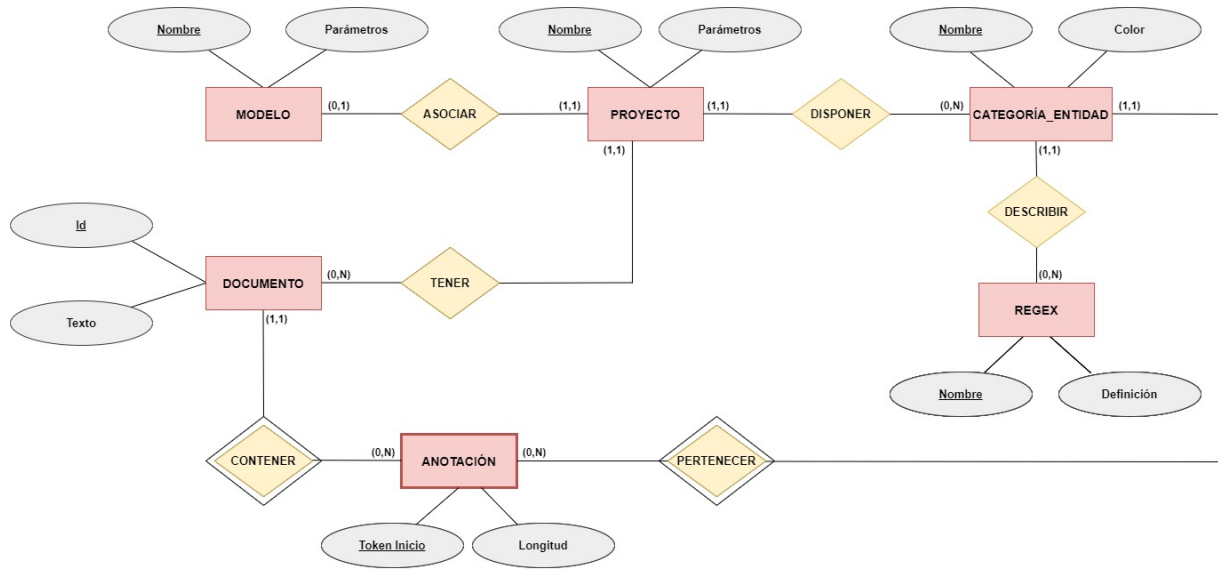


Figura 5.2: Diagrama Entidad-Relación



# Capítulo 6

## Diseño

Una vez realizado el análisis del sistema, se prosigue con la etapa de diseño software.

En la fase de diseño de un producto de software se trata el proceso de planificación de la solución sobre la cual se cimentará la etapa de implementación y posterior codificación del propio código fuente. Su objetivo principal es determinar la estructura general del sistema, identificando las formas de conexión y relación que se producirán entre sus partes. Dentro del diseño se incluye la especificación de las arquitecturas lógica y física del sistema, la construcción del modelo lógico de datos, así como una primera aproximación al diseño de la interfaz de usuario del producto.

### 6.1. Arquitectura Lógica

La arquitectura lógica de un sistema de software define las diferentes partes en las que se estructura el sistema, así como el modo de interrelación entre sus componentes. Proporciona una visión de alto nivel de cómo se encuentra organizado el código del sistema.

La arquitectura lógica involucrada en nuestra aplicación sigue el patrón Modelo-Vista-Template (MVT) de Django, un patrón de diseño derivado del clásico Modelo-Vista-Controlador (MVC). La función de cada componente representado en esta arquitectura se expone a continuación:

- **Controlador:** representa el punto de entrada a la aplicación web, siendo el encargado de realizar la correspondencia (mapeo) entre la URL solicitada en la petición HTTP y su correspondiente vista asociada.

En la arquitectura planteada por Django este elemento se considera dentro del componente Vista, pero en nuestra arquitectura lo presentamos individualmente para una mayor claridad en la explicación.

- **Vista:** representa la capa de lógica de negocio en la que se conecta con el modelo para obtener los datos requeridos y poder entregárselos a una de las plantillas.
- **Modelo:** representa la capa de acceso a datos en la que se interactúa directamente con el componente de almacenamiento de la aplicación para recuperar la información que solicite una de las vistas.

- **Templates (Plantillas):** representa la capa de presentación de datos que retorna al cliente el fichero con los datos aportados por una vista para su posterior renderizado en el navegador. En nuestro caso, son devueltos ficheros HTML, aunque también podrían tratarse otras representaciones como XML, JSON o, simplemente, cadenas de caracteres.
- **Configuración:** representa el componente transversal de configuración de la aplicación. En él se incluyen diferentes ajustes del proyecto como el registro de las aplicaciones, la definición del directorios base para localizar recursos o los dominios asociados a la aplicación web.
- **Almacenamiento:** representa el componente donde se almacena la información de la aplicación. La solución de almacenamiento planteada se basa en el uso de ficheros que guardan la información asociada a cada proyecto de anotación en un directorio particular. Dentro del directorio de cada proyecto se mantiene una estructura definida para almacenar los datos de cada elemento del dominio de la herramienta. Los detalles de implementación concretos pueden consultarse en la Sección 7.2.

Una vez establecidos estos conceptos previos, se presenta un diagrama representando la arquitectura lógica a un alto nivel de abstracción.

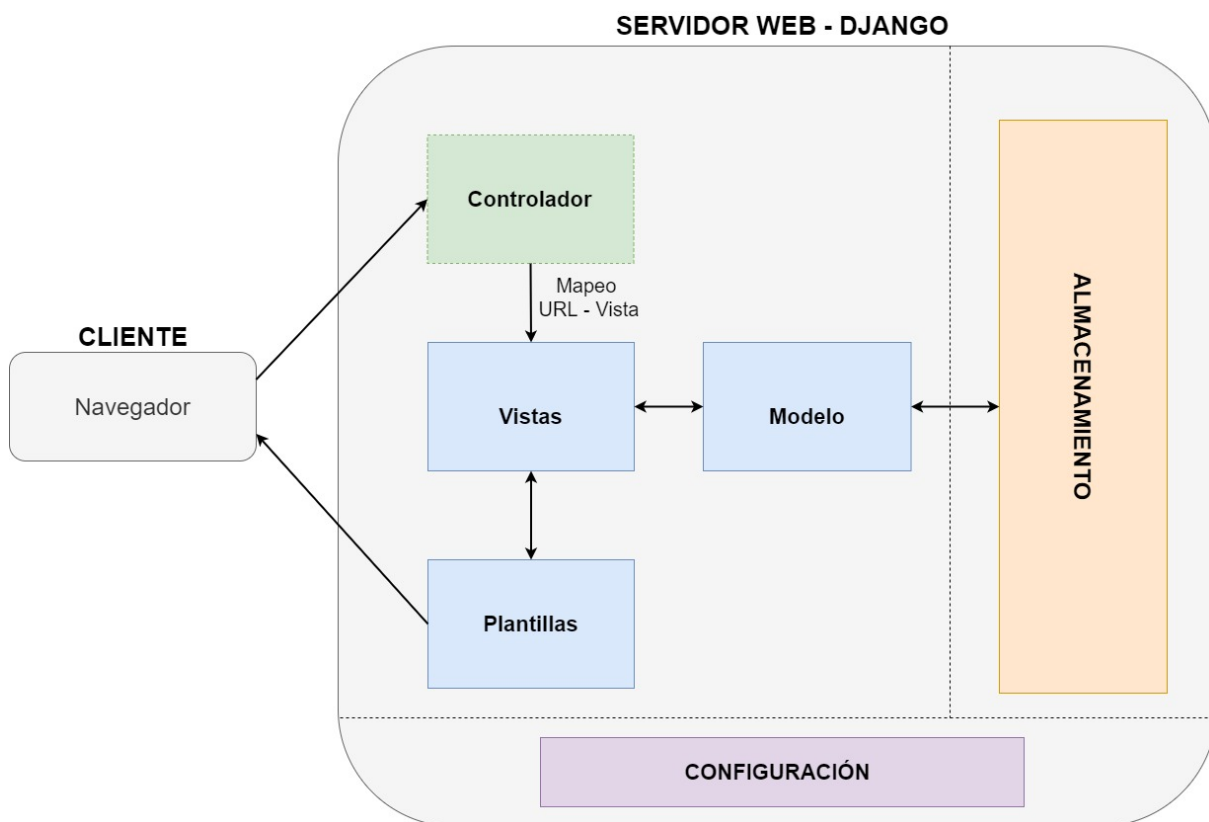


Figura 6.1: Arquitectura Lógica de Alto Nivel

Igualmente, el siguiente diagrama representa la arquitectura lógica del sistema a un mayor nivel de detalle.

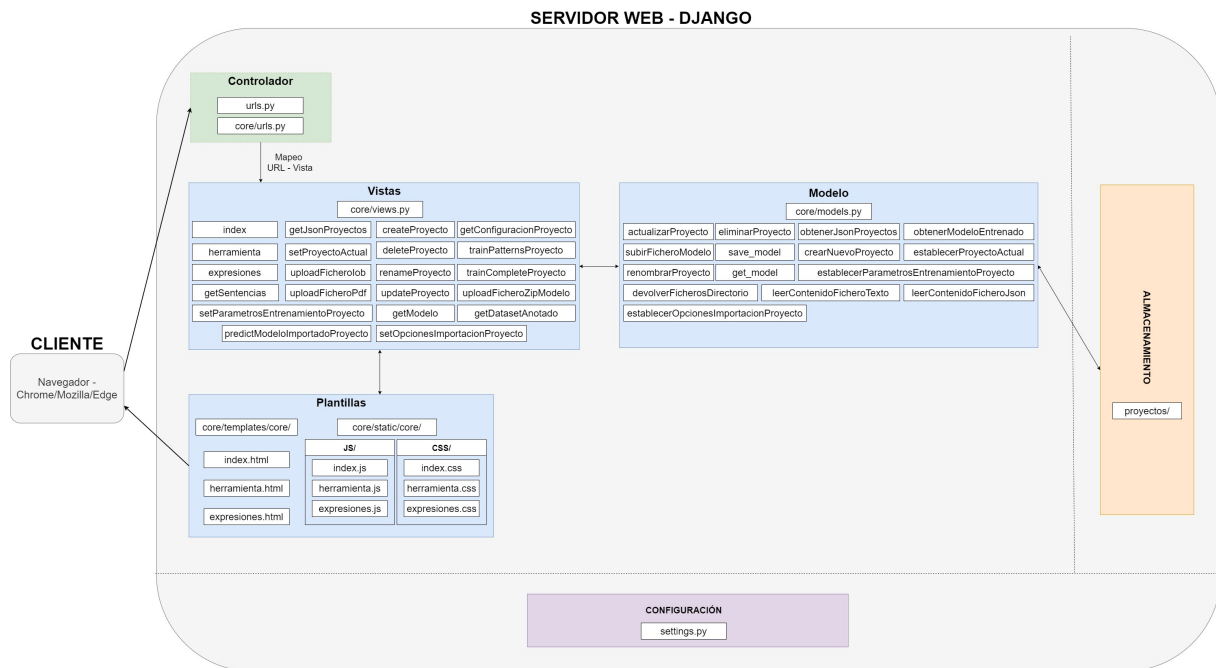


Figura 6.2: Arquitectura Lógica Detallada

Con el objetivo de clarificar el entendimiento del flujo básico de proceso en la aplicación web, se especifican las interacciones que se producen entre los componentes de la aplicación desde la petición HTTP realizado por el cliente hasta la respuesta HTTP devuelta por el servidor.

1. En primer lugar, se recibe en el controlador la petición HTTP enviada desde el cliente y dirigida a una de las URLs definidas en la aplicación. Concretamente, en el script `urls.py`.
2. Seguidamente, se realiza el mapeo entre la URL solicitada y la vista de la aplicación a la que se corresponde.
3. La vista resuelta solicita al modelo los datos requeridos para completar la petición.
4. A continuación, la vista entrega los datos devueltos por el modelo a una de las plantillas definidas.
5. Por último, se retorna como respuesta HTTP la plantilla con los datos convenientemente introducidos para su posterior renderización en el navegador del cliente.

## 6.2. Arquitectura Física

La arquitectura física representa el ambiente físico de implantación de un sistema mostrando los diferentes componentes físicos que intervienen en el flujo de operación del mismo. Describe

la forma de interconexión entre los componentes de hardware involucrados en el proceso para el correcto funcionamiento y despliegue del sistema.

Dentro de los requisitos no funcionales enumerados para nuestro sistema no se han considerado grandes restricciones de seguridad, escalabilidad, fiabilidad o disponibilidad, por lo que la arquitectura física planteada se reduce a una arquitectura básica tipo Cliente-Servidor.

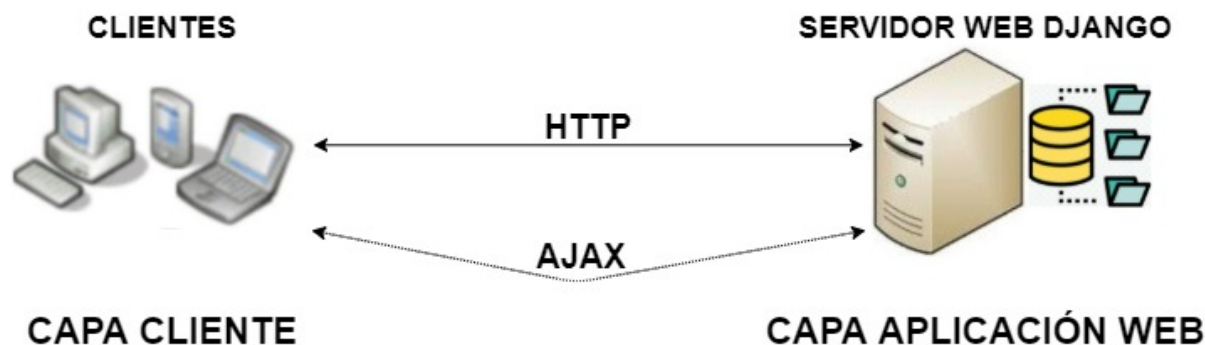


Figura 6.3: Arquitectura Física

### 6.3. Diagrama de Clases

Un diagrama de clases es una herramienta de utilidad para describir la estructura de un sistema a través del modelado de sus entidades fundamentales junto con sus atributos, operaciones y relaciones con otras entidades del sistema. Representa una manera de esquematizar en un simple diagrama la estructura general del sistema.

El diagrama de clases de la Figura 6.4 describe las partes principales que componen nuestra herramienta.

La clase Proyecto actúa como clase principal del sistema englobando al resto de clases, es decir, representa el hecho de que un proyecto de anotación contendrá un modelo de aprendizaje, un conjunto de categorías de entidad y una serie de documentos de texto. Por otra parte, destacan las relaciones de composición existentes entre la clase ANOTACIÓN y las clases DOCUMENTO y CATEGORÍA\_ENTIDAD. Esta relación especifica la necesidad de la existencia de un objeto de una clase en caso de existir un objeto de la otra. En nuestro caso, no puede existir una anotación sin estar contenida en un documento ni sin pertenecer a una categoría de entidad.



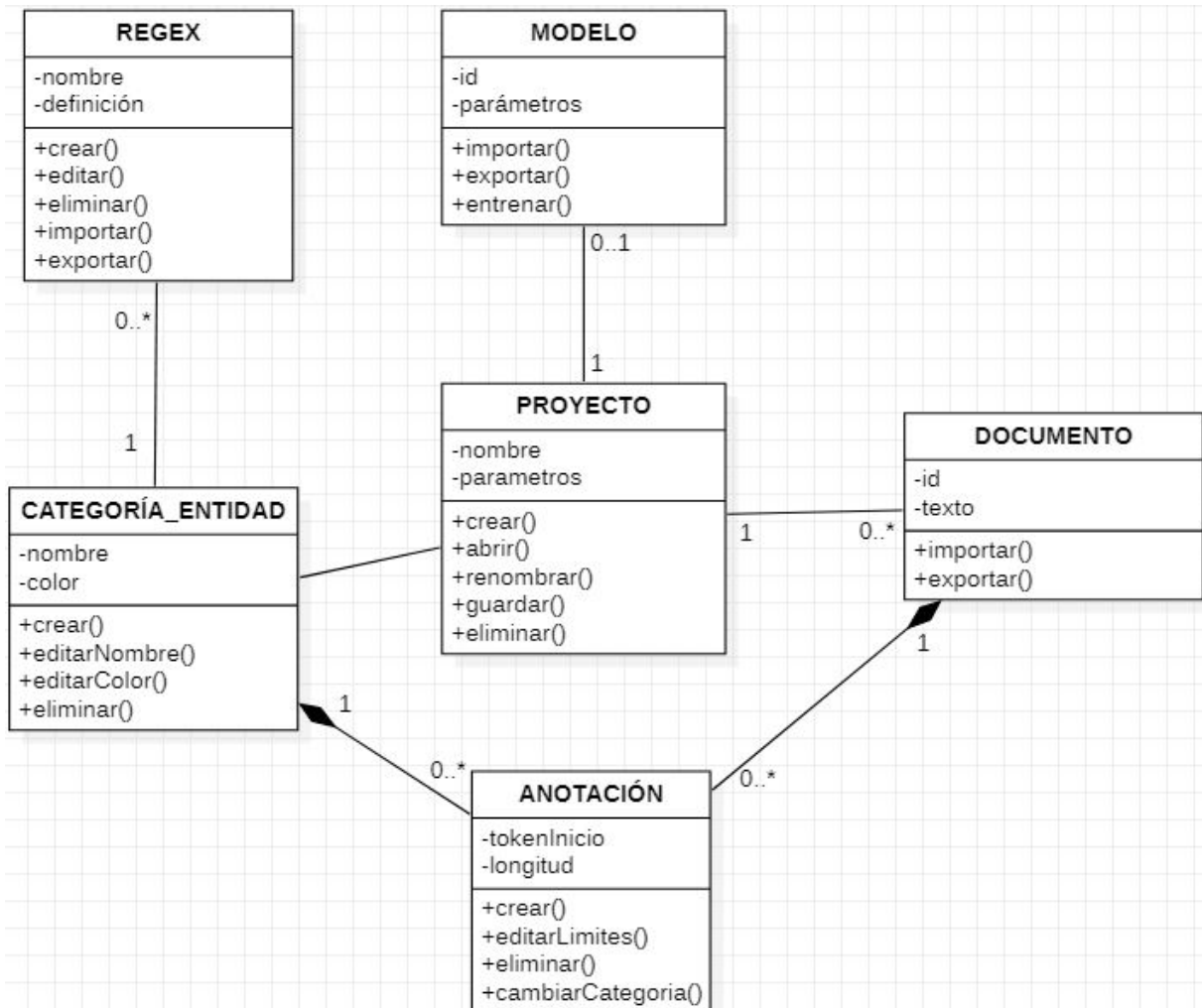


Figura 6.4: Diagrama de Clases

## 6.4. Diagramas de Secuencia

Los diagramas de secuencias nos permiten modelar las interacciones que se producen entre el usuario y los distintos componentes de la aplicación con el fin de llevar a cabo una determinada funcionalidad.

Para este proyecto, se considera de utilidad expresar los diagramas de secuencia relativos a varias de las funcionalidades básicas de la herramienta con el objetivo de comprender su flujo de proceso.

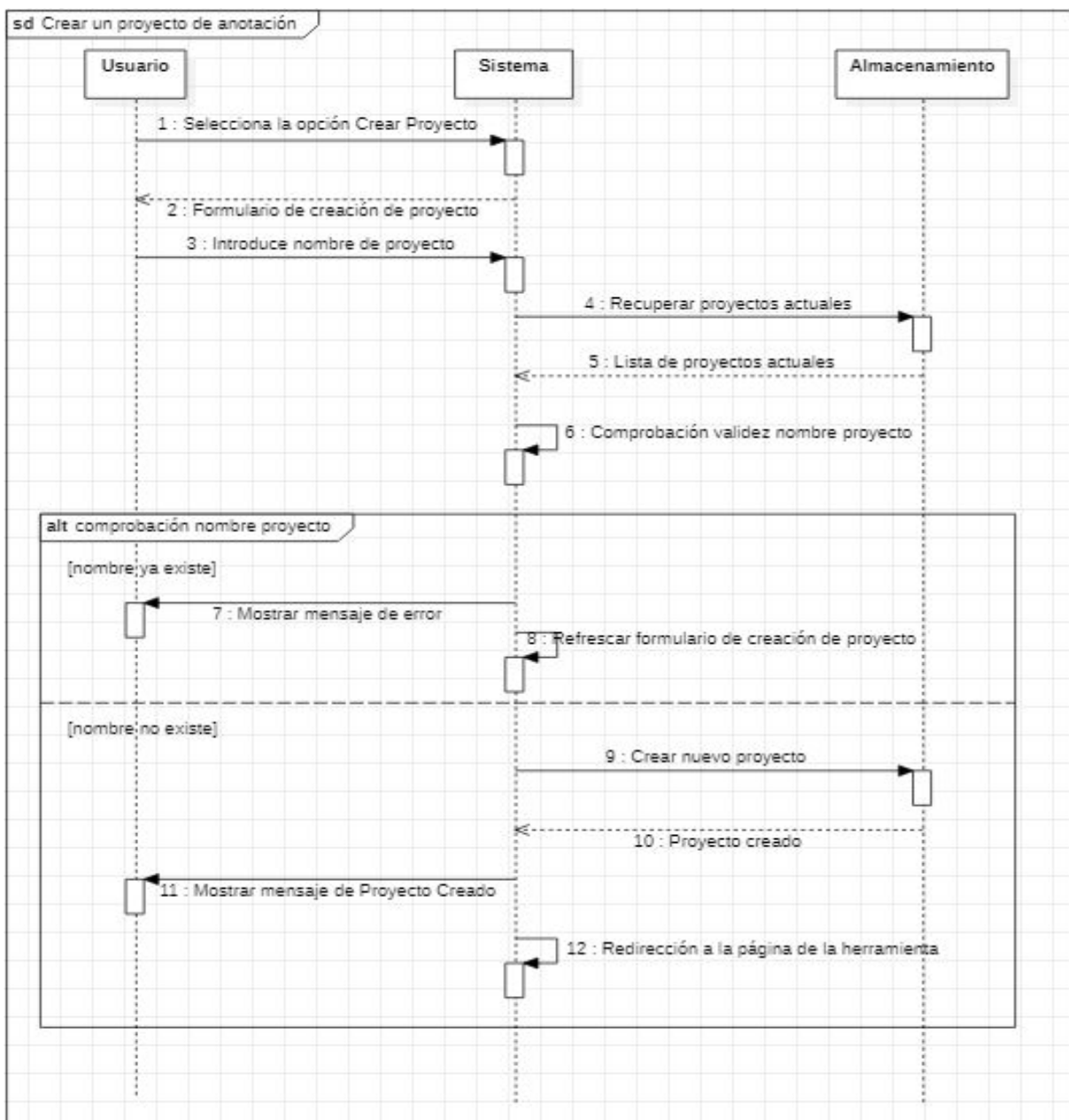


Figura 6.5: Diagrama de Secuencia relativo al Caso de Uso CU-01: Crear un proyecto de anotación.

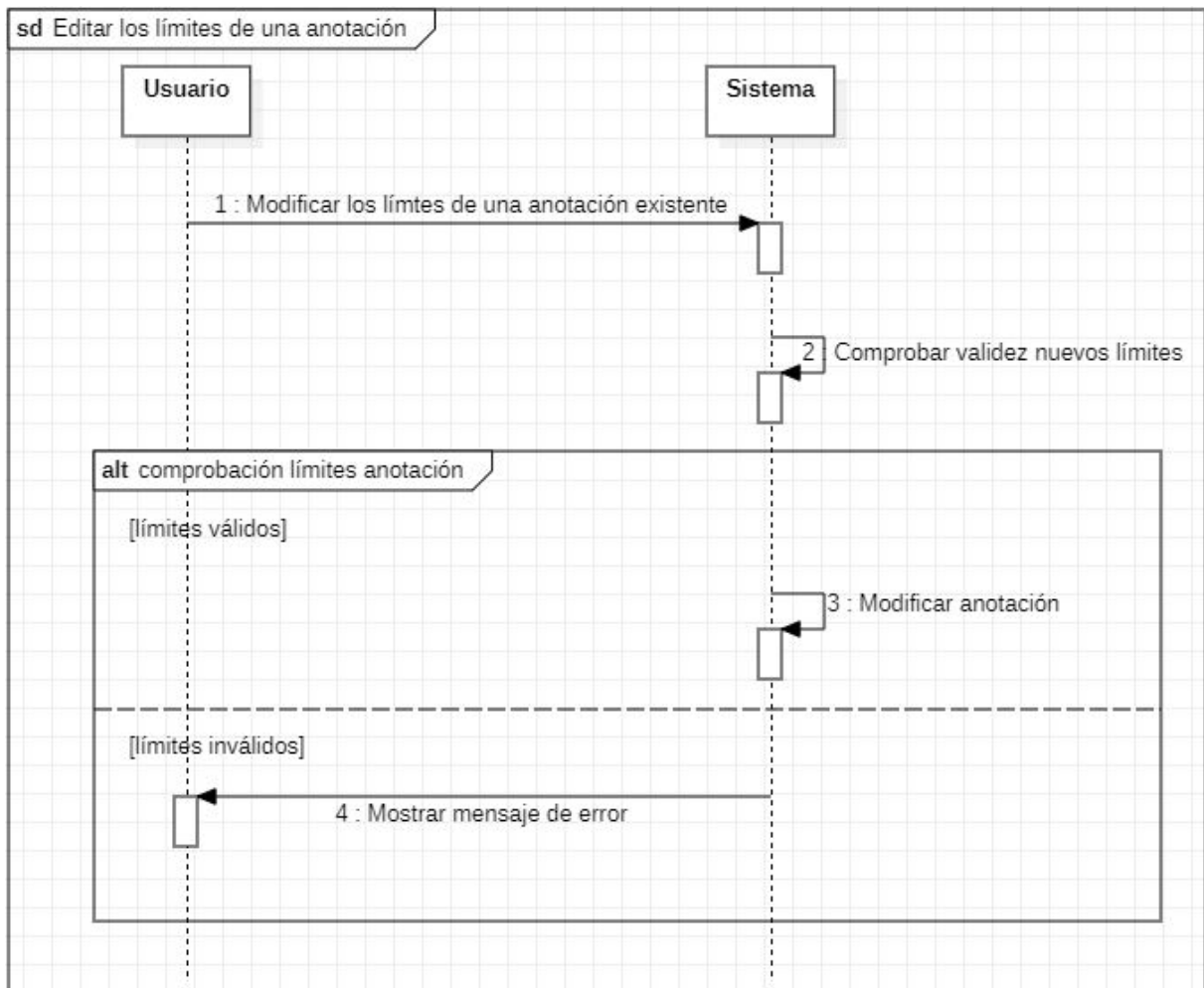


Figura 6.6: Diagrama de Secuencia relativo al Caso de Uso CU-08: Editar los límites de una anotación.

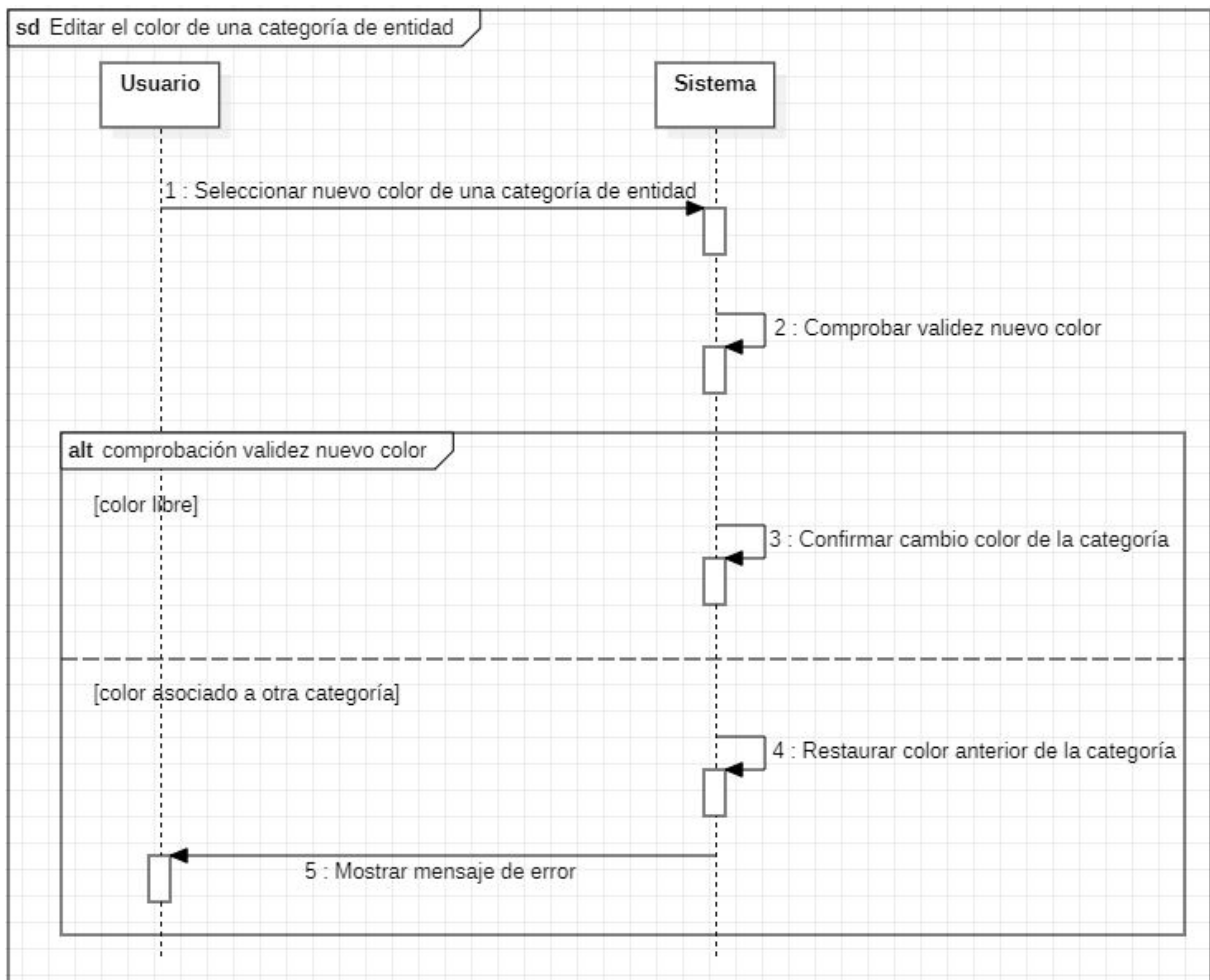


Figura 6.7: Diagrama de Secuencia relativo al Caso de Uso CU-13: Editar el color de una categoría de entidad.

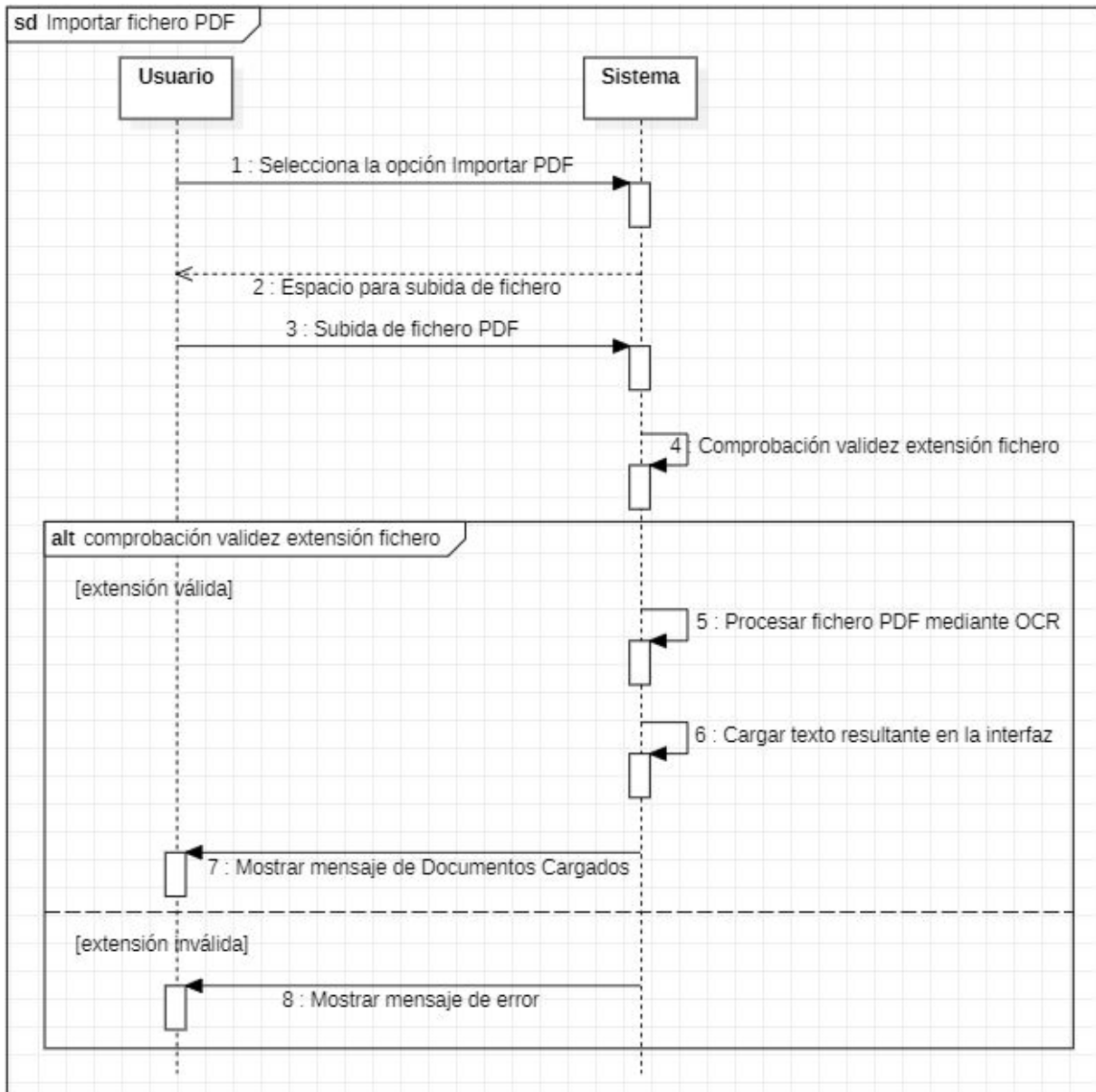


Figura 6.8: Diagrama de Secuencia relativo al Caso de Uso CU-22: Importar fichero PDF.

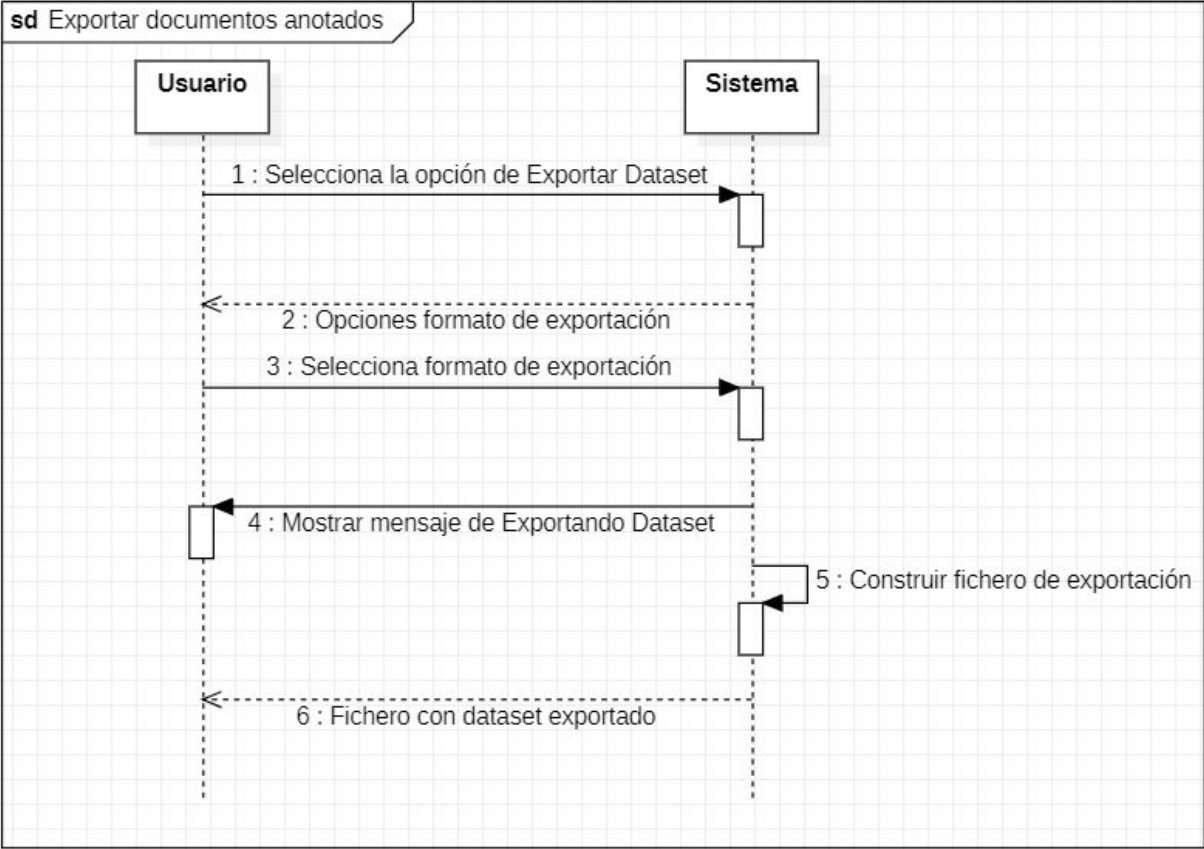


Figura 6.9: Diagrama de Secuencia relativo al Caso de Uso CU-24: Exportar documentos anotados.

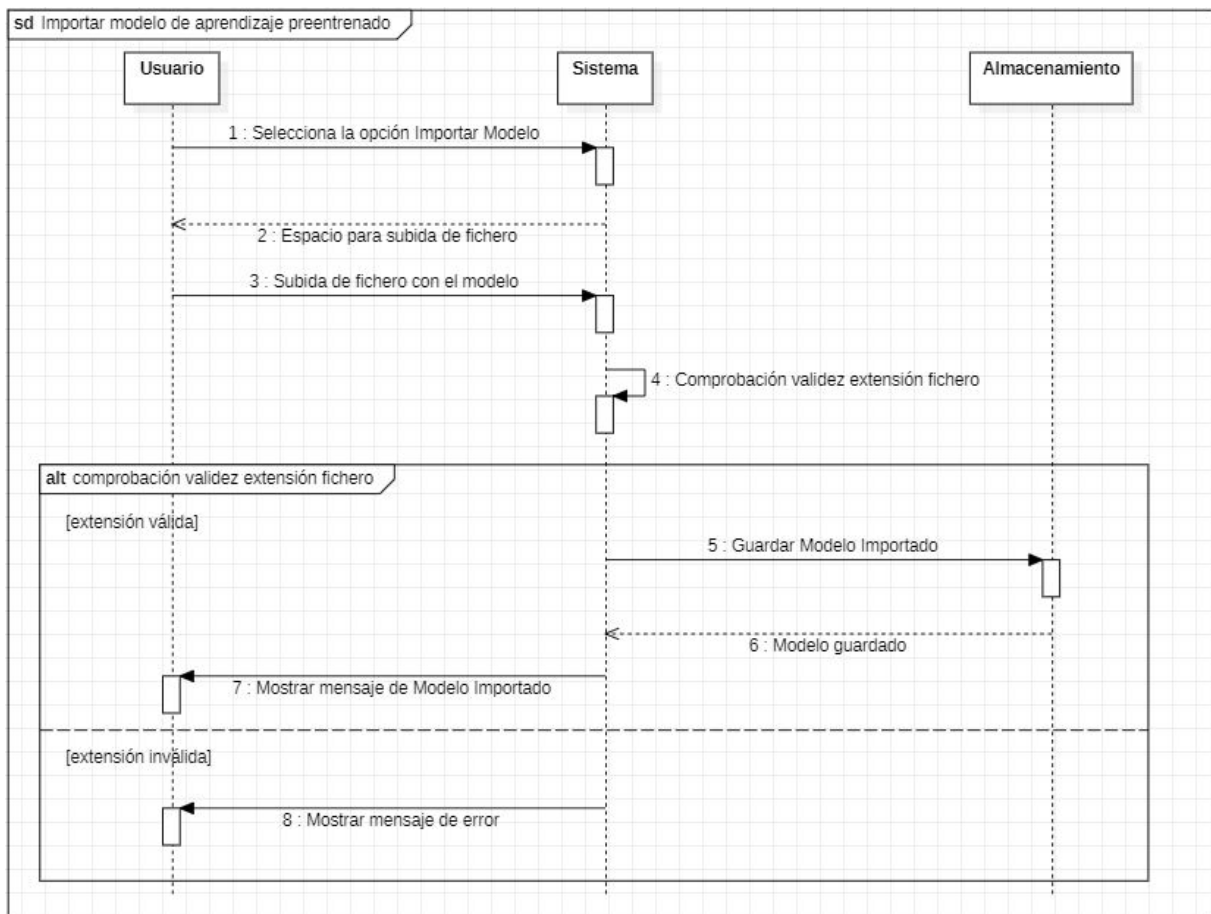


Figura 6.10: Diagrama de Secuencia relativo al Caso de Uso CU-27: Importar modelo de aprendizaje preentrenado.

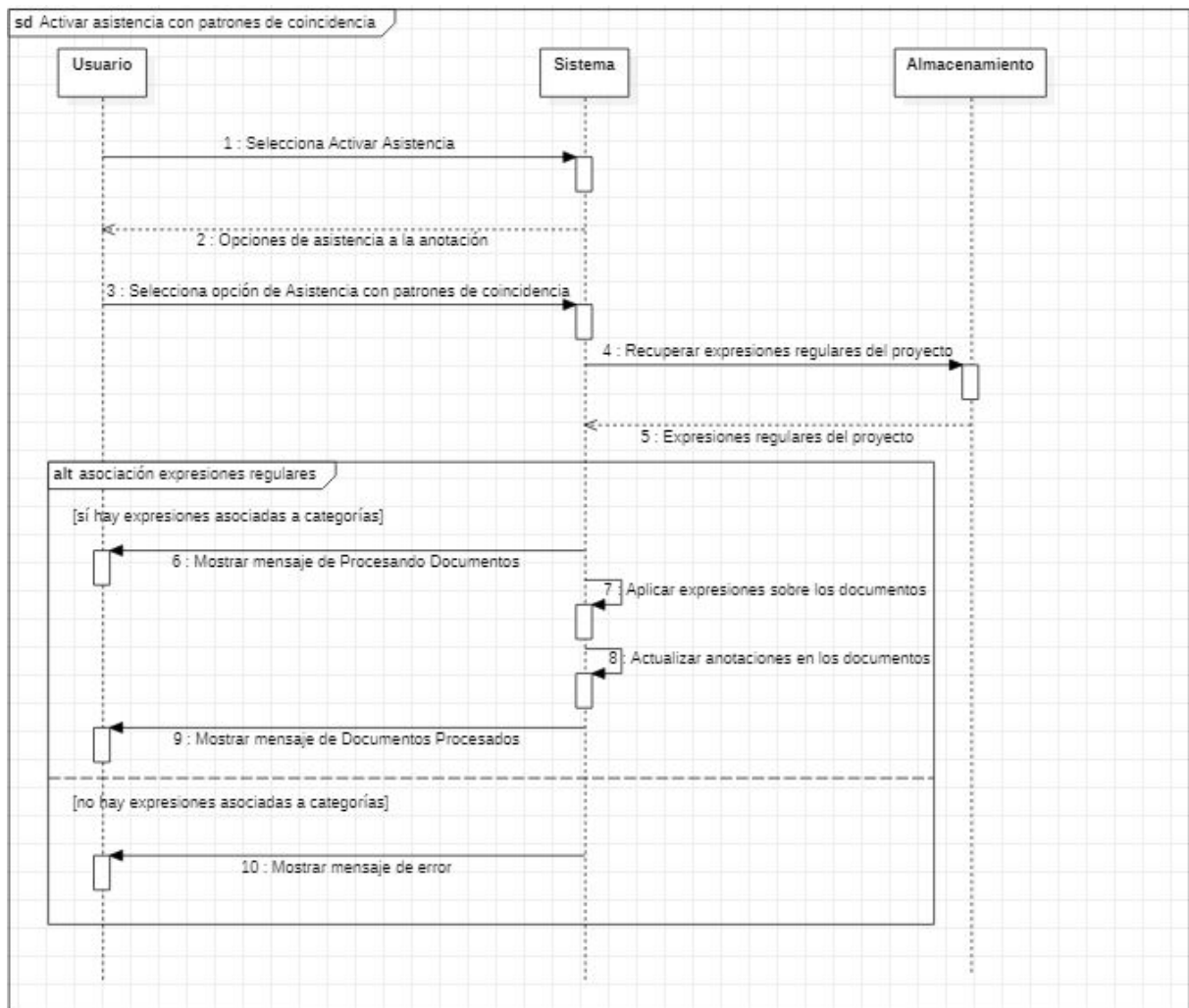


Figura 6.11: Diagrama de Secuencia relativo al Caso de Uso CU-29: Activar asistencia con patrones de coincidencia.



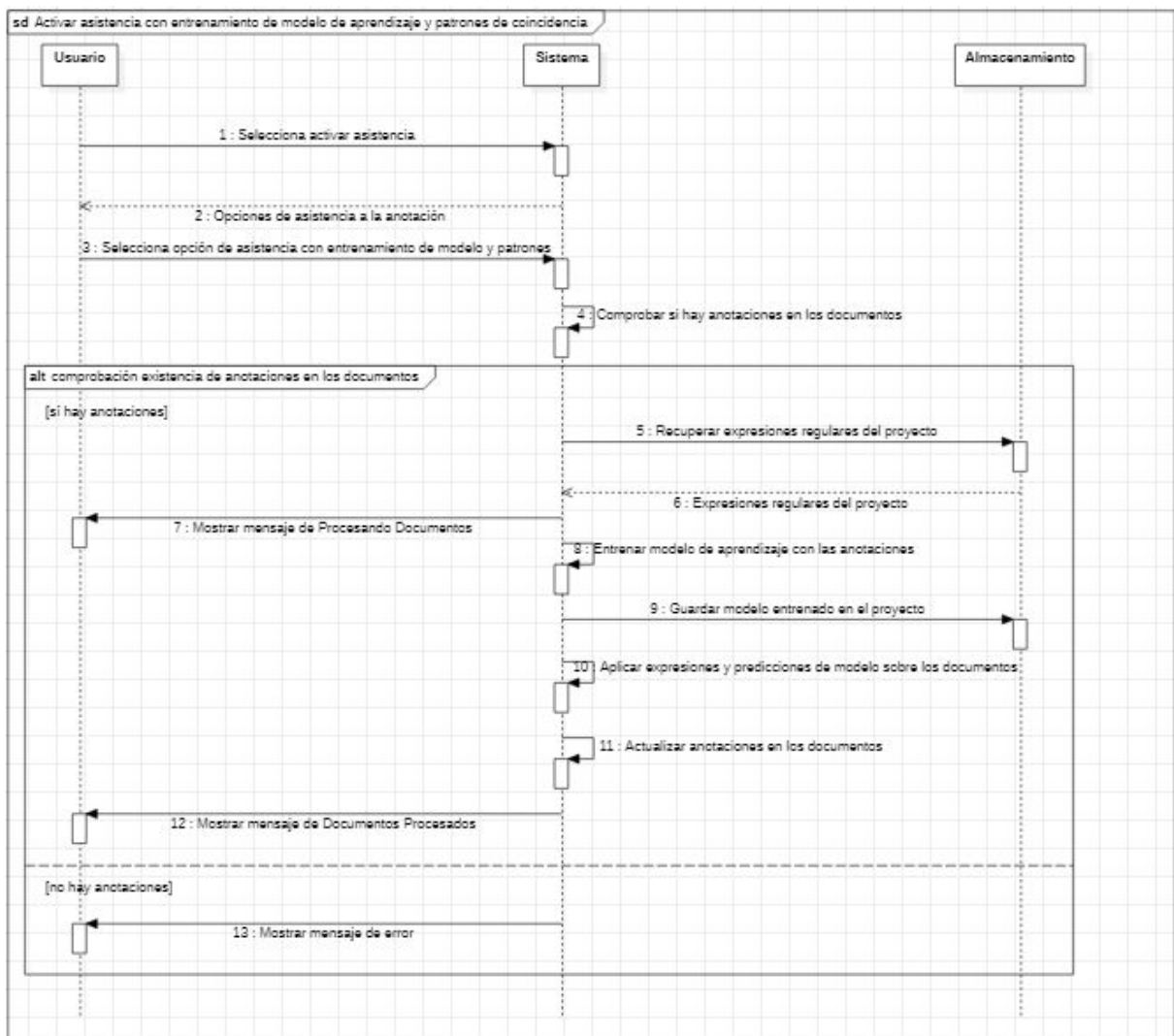


Figura 6.12: Diagrama de Secuencia relativo al Caso de Uso CU-30: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia.

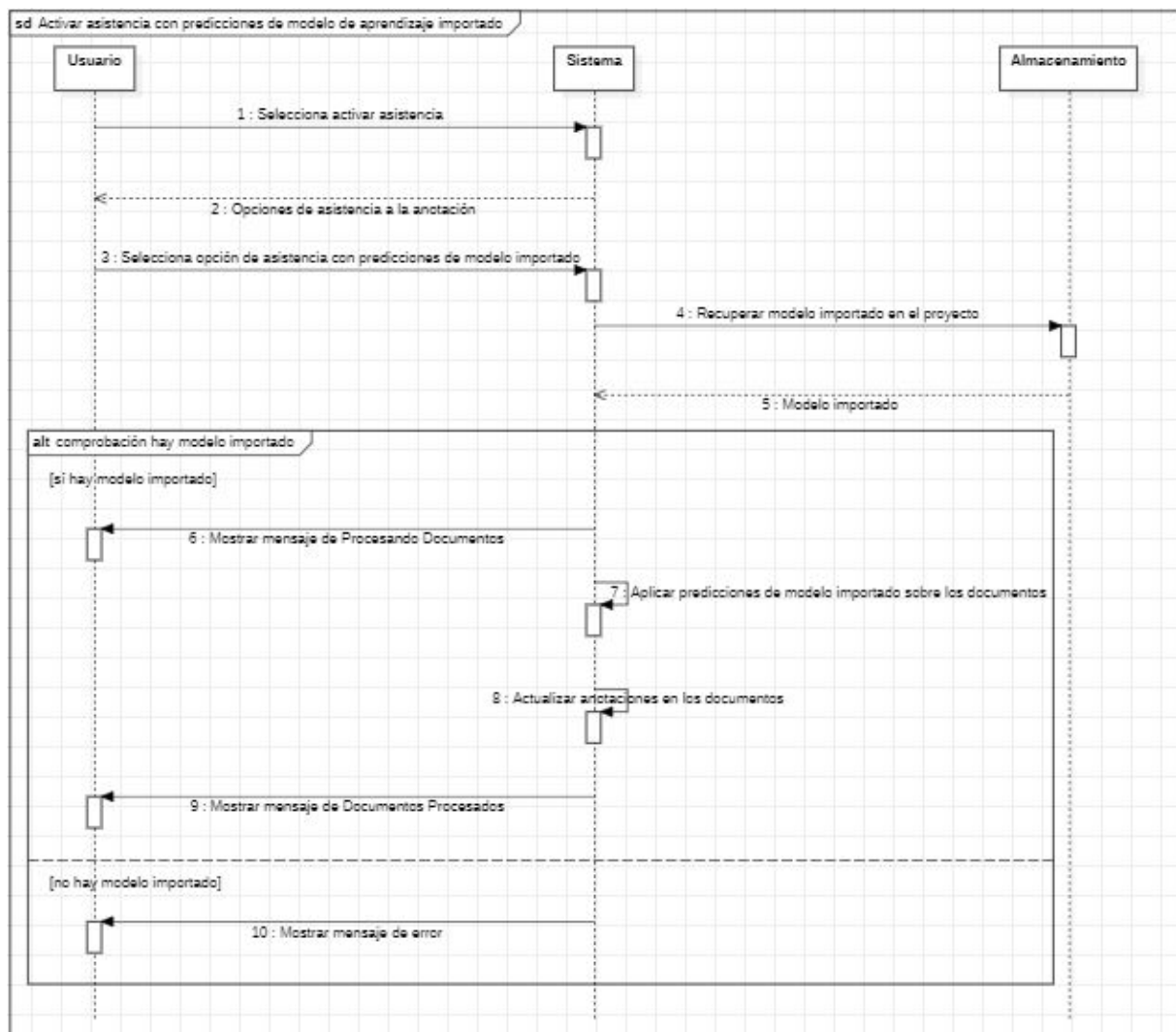


Figura 6.13: Diagrama de Secuencia relativo al Caso de Uso CU-31: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia.

## 6.5. Modelo Lógico de Datos

El modelo lógico de datos es un mecanismo de representación lógica de los elementos que componen el sistema, sin tener en cuenta consideraciones de implementación. Debido a ello, supone una forma útil de poder describir la información independientemente de la solución de almacenamiento que se plantee utilizar.

A continuación, se procede a derivar el modelo lógico relacional a partir del Diagrama Conceptual (Diagrama Entidad-Relación) construido en el capítulo anterior. Véase la Figura 5.2.

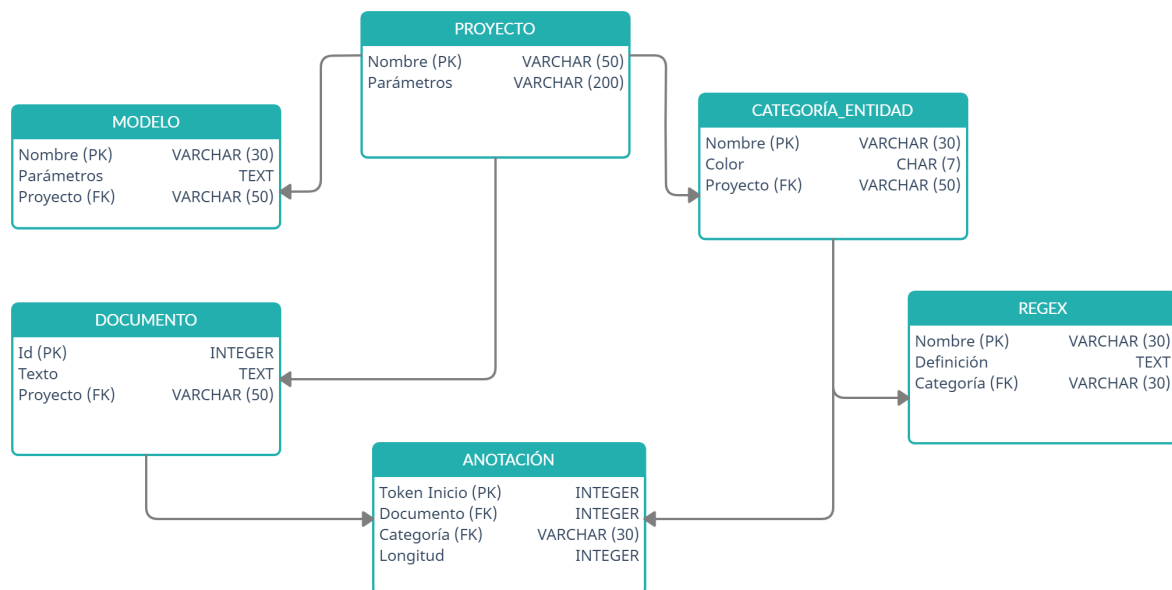


Figura 6.14: Modelo Lógico de Datos

### 6.5.1. Diccionario de Datos

Una vez derivado el modelo lógico de datos del sistema, se continuará especificando el diccionario de datos. Un diccionario de datos es un listado organizado en el que se describe la metainformación asociada a los elementos que forman parte del sistema. Dentro de esta información destacan propiedades relevantes asociadas a cada elemento como son el identificador, el nombre, la definición, el dominio de datos, la unicidad o la posibilidad de ser nulo.

A continuación, se listan los requisitos de información derivados a partir de los modelos conceptual y lógico del sistema.

En primer lugar, se describen los requisitos de información asociados a las entidades.

<b>RI-E01</b>		<b>PROYECTO</b>				
<b>Definición</b>		Un proyecto de anotación representa el contenedor que agrupa todos los componentes de la herramienta.				
<b>ATRIBUTOS</b>						
<b>ID</b>	<b>Nombre</b>	<b>Definición</b>	<b>Dominio</b>	<b>Único</b>	<b>Nulo</b>	<b>Notas</b>
RI-E01.01	Nombre	Nombre identificativo del proyecto	VARCHAR (50)	Sí	No	Actúa como identificador
RI-E01.02	Parámetros	Parámetros de configuración asociados al proyecto	VARCHAR (200)	No	No	-

Tabla 6.1: Requisito de Información RI-E01: PROYECTO

<b>RI-E02</b>		<b>CATEGORÍA_ENTIDAD</b>				
<b>Definición</b>		Una categoría de entidad es el tipo con el que se asigna cada anotación realizada en un texto.				
<b>ATRIBUTOS</b>						
<b>ID</b>	<b>Nombre</b>	<b>Definición</b>	<b>Dominio</b>	<b>Único</b>	<b>Nulo</b>	<b>Notas</b>
RI-E02.01	Nombre	Nombre identificativo de la categoría	VARCHAR (30)	Sí	No	Actúa como identificador
RI-E02.02	Color	Color asociado a la categoría y sus anotaciones	CHAR (7)	Sí	No	-

Tabla 6.2: Requisito de Información RI-E02: CATEGORÍA\_ENTIDAD

<b>RI-E03</b>		<b>DOCUMENTO</b>				
<b>Definición</b>		Un documento representa un texto importado a la herramienta para ser anotado por el usuario.				
<b>ATRIBUTOS</b>						
<b>ID</b>	<b>Nombre</b>	<b>Definición</b>	<b>Dominio</b>	<b>Único</b>	<b>Nulo</b>	<b>Notas</b>
RI-E03.01	Id	Identificativo asociado al documento	INTEGER	Sí	No	Actúa como identificador
RI-E03.02	Texto	Texto a anotar del documento	TEXT	No	No	-

Tabla 6.3: Requisito de Información RI-E03: DOCUMENTO

<b>RI-E04</b>		<b>MODELO</b>				
<b>Definición</b>		Un modelo representa el modelo de aprendizaje entrenado en el proyecto para extraer entidades nombradas en textos.				
<b>ATRIBUTOS</b>						
<b>ID</b>	<b>Nombre</b>	<b>Definición</b>	<b>Dominio</b>	<b>Único</b>	<b>Nulo</b>	<b>Notas</b>
RI-E04.01	Nombre	Nombre identificativo del modelo	VARCHAR (30)	Sí	No	Actúa como identificador
RI-E04.02	Parámetros	Parámetros propios del modelo de aprendizaje	TEXT	No	No	-

Tabla 6.4: Requisito de Información RI-E04: MODELO

<b>RI-E05</b>		<b>REGEX</b>				
<b>Definición</b>		Una regex (expresión regular) es una secuencia de caracteres que conforman un patrón de búsqueda sobre el texto. Cada regex es asociada a una categoría de entidad para conformar un patrón con el que buscar coincidencias en los textos.				
<b>ATRIBUTOS</b>						
<b>ID</b>	<b>Nombre</b>	<b>Definición</b>	<b>Dominio</b>	<b>Único</b>	<b>Nulo</b>	<b>Notas</b>
RI-E04.01	Nombre	Nombre identificativo de la regex	VARCHAR (30)	Sí	No	Actúa como identificador
RI-E04.02	Definición	Secuencia de caracteres que forman la regex	TEXT	Sí	No	-

Tabla 6.5: Requisito de Información RI-E05: REGEX

<b>RI-E06</b>		<b>ANOTACIÓN</b>				
<b>Definición</b>		Una anotación representa una secuencia de 1 o más tokens del texto etiquetados con alguna de las categorías de entidad definidas en el proyecto. La existencia de una anotación está supedita a la existencia del documento y la categoría de entidad a los que está asociada.				
<b>ATRIBUTOS</b>						
<b>ID</b>	<b>Nombre</b>	<b>Definición</b>	<b>Dominio</b>	<b>Único</b>	<b>Nulo</b>	<b>Notas</b>
RI-E04.01	Token inicio	Offset dentro del documento donde empieza la anotación	INTEGER	Sí	No	Actúa como identificador
RI-E04.02	Longitud	Número de tokens que componen la anotación	INTEGER	No	No	-

Tabla 6.6: Requisito de Información RI-E06: ANOTACIÓN

Por otra parte, se especifican los requisitos de información asociados a las relaciones existentes.

<b>RI-R01</b>	<b>DISPONER</b>		
<b>Definición</b>	La relación DISPONER describe la asociación entre un proyecto y las categorías de entidad definidas en él.		
<b>ENTIDADES</b>			
<b>ID</b>	<b>Nombre</b>	<b>Participación</b>	<b>Cardinalidad</b>
RI-E01	PROYECTO	0	N
RI-E02	CATEGORÍA_ENTIDAD	1	1

Tabla 6.7: Requisito de Información RI-R01: DISPONER

<b>RI-R02</b>	<b>TENER</b>		
<b>Definición</b>	La relación TENER describe la asociación entre un proyecto y los documentos importados dentro de él.		
<b>ENTIDADES</b>			
<b>ID</b>	<b>Nombre</b>	<b>Participación</b>	<b>Cardinalidad</b>
RI-E01	PROYECTO	0	N
RI-E03	DOCUMENTO	1	1

Tabla 6.8: Requisito de Información RI-R02: TENER

<b>RI-R03</b>	<b>ASOCIAR</b>		
<b>Definición</b>	La relación ASOCIAR describe la asociación entre un proyecto y el modelo de aprendizaje entrenado dentro de él.		
<b>ENTIDADES</b>			
<b>ID</b>	<b>Nombre</b>	<b>Participación</b>	<b>Cardinalidad</b>
RI-E01	PROYECTO	0	1
RI-E04	MODELO	1	1

Tabla 6.9: Requisito de Información RI-R03: ASOCIAR

<b>RI-R04</b>	<b>DESCRIBIR</b>		
<b>Definición</b>	La relación DESCRIBIR hace referencia a la asociación entre una expresión regular y la categoría de entidad a la que está asignada.		
<b>ENTIDADES</b>			
<b>ID</b>	<b>Nombre</b>	<b>Participación</b>	<b>Cardinalidad</b>
RI-E02	CATEGORÍA_ENTIDAD	0	N
RI-E05	REGEX	1	1

Tabla 6.10: Requisito de Información RI-R04: DESCRIBIR

<b>RI-R05</b>	<b>CONTENER</b>		
<b>Definición</b>	La relación CONTENER describe la dependencia entre una anotación y el documento en el que se realiza.		
<b>ENTIDADES</b>			
<b>ID</b>	<b>Nombre</b>	<b>Participación</b>	<b>Cardinalidad</b>
RI-E03	DOCUMENTO	0	N
RI-E06	ANOTACIÓN	1	1

Tabla 6.11: Requisito de Información RI-R05: CONTENER

<b>RI-R06</b>	<b>PERTENECER</b>		
<b>Definición</b>	La relación PERTENECER hace referencia a la dependencia entre una anotación y la categoría de entidad de la que forma parte.		
<b>ENTIDADES</b>			
<b>ID</b>	<b>Nombre</b>	<b>Participación</b>	<b>Cardinalidad</b>
RI-E02	CATEGORÍA_ENTIDAD	0	N
RI-E06	ANOTACIÓN	1	1

Tabla 6.12: Requisito de Información RI-R06: PERTENECER



## 6.6. Diseño de Interfaz

El diseño de interfaz pretende acometer una primera aproximación al diseño de la interfaz de usuario de la aplicación. Trata las consideraciones relacionadas con la interfaz de usuario que se le presentará al usuario de la herramienta. Dentro de estas consideraciones se incluyen una descripción, condiciones de activación, eventos realizables y un boceto de alto nivel para cada una de las vistas que compondrán la interfaz de usuario.

A continuación, se presentan las tablas realizadas para el diseño de interfaz sobre las páginas que van a formar la herramienta. Estas páginas son, básicamente, una página de inicio de acceso a proyectos, una página principal de la herramienta para la anotación en el proyecto y otra página para la gestión de expresiones regulares en el proyecto.

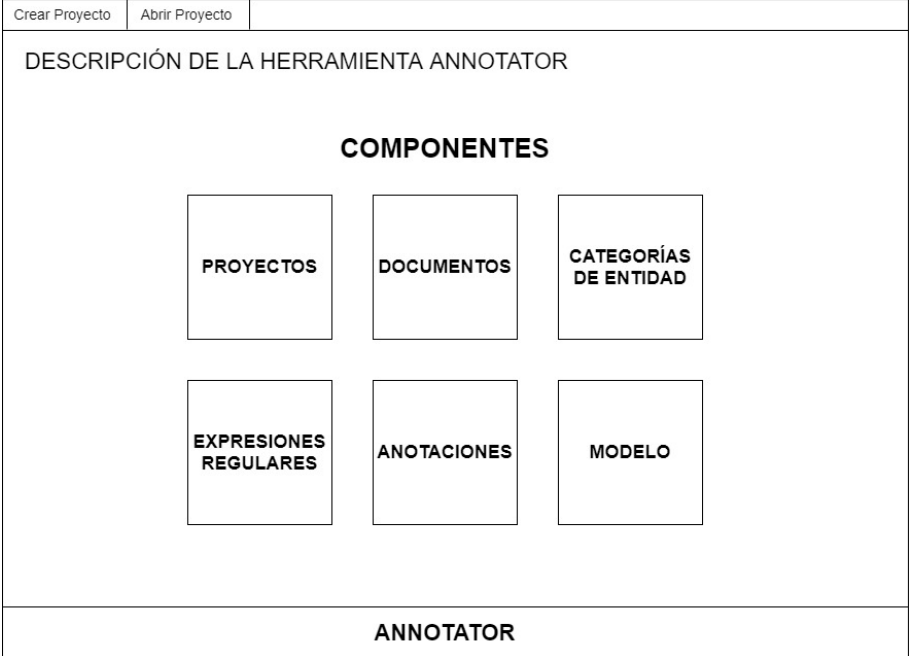
DI-01	<b>Página de Inicio (inicio.html)</b>
Descripción	Esta es la página de inicio de la herramienta en la que se crean nuevos proyectos y se abren los existentes. También, ofrece una descripción de la herramienta y sus componentes.
Activación	Al acceder a la herramienta o a través de una opción de menú de las otras páginas.
Boceto	
Eventos	Crear un nuevo proyecto o abrir uno existente.

Tabla 6.13: Diseño de Interfaz DI-01: Página de Inicio

<b>DI-02</b>	<b>Página Principal de la Herramienta (herramienta.html)</b>
Descripción	Esta es la página principal de la herramienta donde se llevan a cabo la mayoría de sus funcionalidades.
Activación	Al crear o abrir un proyecto o a través de una opción de menú de la Página de Gestión de Expresiones.
Boceto	
Eventos	Importar nuevos documentos, importar modelo de aprendizaje, gestionar categorías de entidad, gestionar anotaciones, activar asistencia a la anotación, exportar dataset de documentos anotados, exportar modelo entrenado y visualizar información, entre otras.

Tabla 6.14: Diseño de Interfaz DI-02: Página Principal de la Herramienta

<b>DI-01</b>	<b>Página de Gestión de Expresiones Regulares (expresiones.html)</b>																				
Descripción	Esta es la página de gestión de las expresiones regulares vinculadas a un proyecto. Permite crear nuevas expresiones regulares y asociarlas a categorías de entidad del proyecto.																				
Activación	A través de una opción de menú de la Página Principal de la Herramienta.																				
Boceto	<p><b>Definición de Expresiones Regulares</b></p> <p>REGEX DNI <input type="text" value="[0-9]{8}[A-Za-z]"/></p> <p>REGEX E-MAIL <input type="text" value="((?&lt;=&amp;#x27;)(?&lt;=&amp;#x27;) [w-]+(\.[w-]+)*@[w-]+\.+lw+)(?=&amp;#x27; s\.)"/></p> <p>REGEX TELÉFONO <input type="text" value="((?&lt;=&amp;#x27;)(?&lt;=&amp;#x27;) 6789 (((\d{2}(?P&lt;sep&gt;[- ]?)d{2}))(\d{1}(?P&lt;sep&gt;[- ]?)d{1})))"/></p> <p><b>Asignación de Expresiones Regulares</b></p> <table border="1"> <thead> <tr> <th></th> <th>REGEX DNI</th> <th>REGEX E-MAIL</th> <th>REGEX TELÉFONO</th> </tr> </thead> <tbody> <tr> <td>DNI</td> <td><input checked="" type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> <tr> <td>E-MAIL</td> <td><input type="radio"/></td> <td><input checked="" type="radio"/></td> <td><input type="radio"/></td> </tr> <tr> <td>DIRECCIÓN</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> <tr> <td>TELÉFONO</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input checked="" type="radio"/></td> </tr> </tbody> </table> <p><b>ENTIDADES</b></p> <p>Añadir Nueva Categoría</p> <ul style="list-style-type: none"> <li>DNI</li> <li>E-MAIL</li> <li>DIRECCIÓN</li> <li>TELÉFONO</li> </ul> <p><b>ANNOTATOR</b></p>		REGEX DNI	REGEX E-MAIL	REGEX TELÉFONO	DNI	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	E-MAIL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	DIRECCIÓN	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	TELÉFONO	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
	REGEX DNI	REGEX E-MAIL	REGEX TELÉFONO																		
DNI	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>																		
E-MAIL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>																		
DIRECCIÓN	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																		
TELÉFONO	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>																		
Eventos	Gestionar expresiones regulares y asociarlas a categorías de entidad.																				

Tabla 6.15: Diseño de Interfaz DI-03: Página de Gestión de Expresiones Regulares



# Capítulo 7

## Implementación

La implementación del sistema comprende el proceso de materialización de los artefactos derivados de la etapa de diseño en el código funcional del sistema.

En esta sección del proyecto se pone en contexto el entorno en que se ha desarrollado el producto y se describen las herramientas y tecnologías involucradas en el proceso de implementación. A su vez, se detallará la organización del código de la herramienta y se mostrarán ciertos fragmentos del propio código fuente que sean de interés.

### 7.1. Descripción Técnica

En este proyecto se ha trabajado con una metodología ágil en la que al final de cada Sprint se ha entregado un incremento de producto totalmente funcional con parte de la funcionalidad proyectada en el alcance del proyecto terminada. Por lo tanto, el código fuente de la herramienta ha sido implementado de manera incremental a lo largo de los sprints, siguiendo los preceptos del desarrollo ágil.

Los datos con los que se ha trabajado en el proyecto han proporcionado un caso de uso real sobre el que probar la herramienta implementada. Estos datos pertenecen al Ayuntamiento de Valladolid y provienen de un estudio realizado por el grupo de investigación dataAi de la Universidad de Valladolid en colaboración con el Ayuntamiento de Valladolid. Para su obtención se firmó un Acuerdo de Secreto y Confidencialidad, por lo que no se hará alusión a su contenido en esta memoria.

En cuanto a la funcionalidad implementada, la herramienta Annotator cuenta con toda la funcionalidad prevista en el alcance del proyecto.

El árbol de características simplificado que se construyó para plantear el alcance del producto aparece en la Figura 7.1.

Este diagrama se estructura en 3 ramas principales, las cuales describen cada una de las partes a implementar en la herramienta.

- **Entrada y Salida de Datos**

Dentro de esta rama se trata todo lo relativo a la importación y exportación de datos de la herramienta.

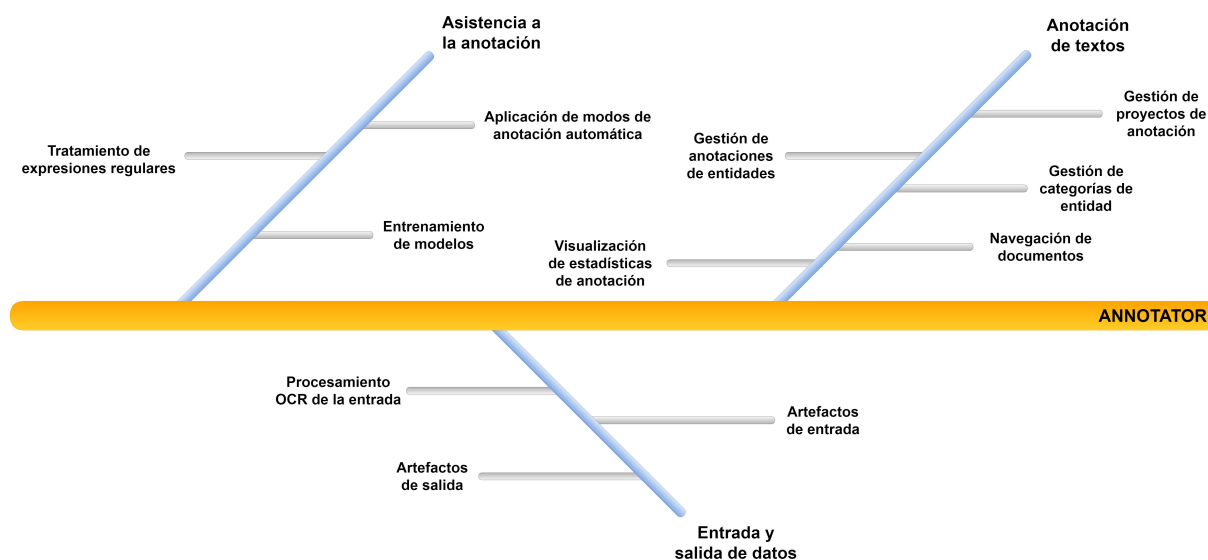


Figura 7.1: Árbol de Características Simplificado

- **Artefactos de entrada:** representa la entrada importada por el usuario a la herramienta. Incluye la importación de nuevos documentos de texto, datasets anotados, ficheros de expresiones regulares y modelo de aprendizaje preentrenado.
- **Artefactos de salida:** representa la salida exportada por la herramienta a demanda del usuario. Incluye la exportación del dataset de documentos anotados, el modelo de aprendizaje entrenado y el fichero de expresiones regulares.
- **Procesamiento OCR de la entrada:** representa el proceso OCR realizado sobre los ficheros PDF de entrada para extraer el texto que contienen. Incluye la importación de los PDFs así como la obtención de su texto y posterior carga en la herramienta.

#### ■ Anotación de textos

Dentro de este bloque se consideran todas las características asociadas con la capacidad fundamental de anotación de textos.

- **Gestión de proyectos de anotación:** representa la gestión de los proyectos de anotación, el elemento que actúa como contenedor del resto de componentes del sistema. Incluye la creación y eliminación de proyectos por parte del usuario, así como su renombramiento y almacenamiento.
- **Gestión de categorías de entidad:** representa la gestión de los tipos de entidades con los que realizar las anotaciones en los documentos. Incluye la creación, eliminación y edición de las categorías de entidad.
- **Gestión de anotaciones de entidades:** representa la gestión de las anotaciones de entidades nombradas dentro de los documentos. Incluye la creación, eliminación y modificación de los límites y categoría de entidad de las anotaciones. A su vez,

- **Navegación de documentos:** representa la característica de navegación entre los documentos de entrada importados a la herramienta. Así mismo, se considera dentro de esta subrama la propia navegación entre las sentencias de un mismo documento.
- **Visualización de estadísticas de anotación:** representa la posibilidad por parte del usuario de la herramienta de visualizar estadísticas del proyecto de anotación actual. Incluye el número de documentos importados, de categorías de entidad existentes y de anotaciones realizadas.

#### ■ Asistencia a la anotación

Dentro de esta rama se tienen en cuenta todas las características asociadas a la asistencia a la anotación.

- **Aplicación de modos de anotación automática:** representa las distintas opciones de asistencia a la anotación que el usuario de la herramienta puede activar a demanda. Incluye 3 modos diferentes de asistencia:
  1. Asistencia con patrones de coincidencia
  2. Asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia
  3. Asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia
- **Entrenamiento de modelos:** representa la parte del entrenamiento de un nuevo modelo de aprendizaje automático a partir de las anotaciones existentes para la predicción de nuevas entidades nombradas en los documentos del proyecto. Incluye también la configuración de parámetros del entrenamiento y la visualización de los resultados obtenidos en el mismo.
- **Tratamiento de expresiones regulares:** representa la gestión de las expresiones regulares definidas en el proyecto. Esta gestión abarca desde la creación, eliminación y edición de las propias expresiones regulares hasta su asociación con categorías de entidad del proyecto. Dichas asociaciones entre expresiones y categorías conforman los patrones de coincidencia utilizados dentro de los modos de asistencia a la anotación.

El Árbol de Características completo asociado a la herramienta puede observarse en la Figura 1.2.

De cara a acometer la implementación que diera soporte a toda la funcionalidad planteada para nuestra herramienta, se propuso como solución el pipeline de procesamiento de la Figura 7.2.

En él aparece representado el flujo de proceso completo involucrado en la herramienta, abarcando desde la entrada de los artefactos a la herramienta por parte del usuario hasta la exportación de los artefactos de salida una vez completado el proceso de anotación sobre la entrada.

A continuación, se procede describir cada una de las partes que componen este pipeline de procesamiento para la anotación de entidades en la herramienta.

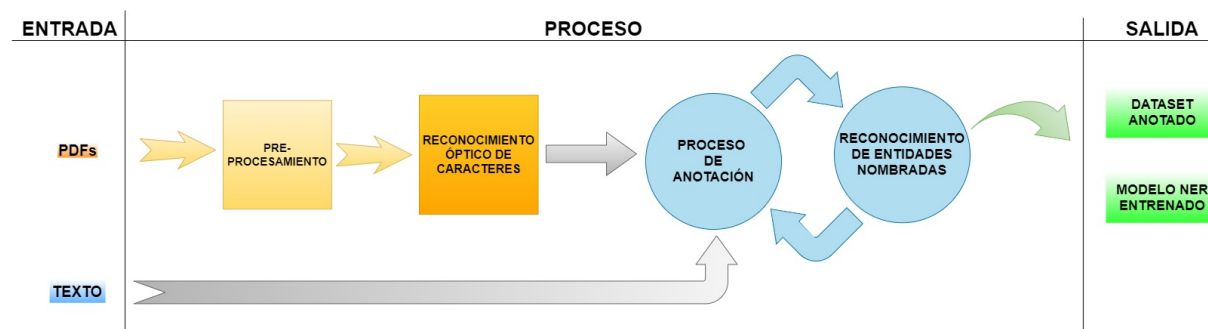


Figura 7.2: Pipeline de Procesamiento de Annotator

■ **Entrada**

Representa los documentos que importa el usuario a la herramienta. Éstos pueden ser directamente ficheros de texto plano, que no requieren un preprocesamiento antes de ser cargados en la herramienta, o ficheros PDF, que deben ser preprocesados antes de su carga en la herramienta.

■ **Proceso**

Representa el proceso completo que tiene lugar entre la importación y la exportación de datos de la herramienta.

• **Preprocesamiento:**

En primer lugar, se incluye una etapa de preprocesamiento en caso de que la entrada sea un fichero PDF.

Dentro de dicho procesamiento se lleva a cabo el recorte de las páginas del fichero PDF con el objetivo de acotar el espacio dentro de cada página donde se encuentra el texto a extraer.

Para ello, el usuario puede establecer porcentajes de recorte de páginas dentro de las opciones de importación del proyecto. En nuestro caso de uso particular, los ficheros PDF importados a la herramienta contenían metainformación en los márgenes de las páginas que enmarcaban el contenido real. Para la obtención de dicho contenido deseado se aplicaron porcentajes de recorte en las páginas para eliminar su metainformación.

• **Reconocimiento Óptico de Caracteres:**

Igualmente, esta etapa de Reconocimiento Óptico de Caracteres solo se llevará a cabo en caso de tratarse con un fichero PDF.

Tras acotar la sección del contenido de las páginas del PDF, se convierten éstas en imágenes para poder llevar a cabo el procesamiento OCR sobre ellas.

El motor OCR utilizado, Pytesseract, procesará cada imagen de página y extraerá su texto contenido.



Una vez obtenido el texto completo, se aplicarán expresiones regulares con las que descartar espacios sobrantes, saltos de líneas u otros caracteres especiales como medio para la limpieza del texto.

Finalmente, el texto resultante será cargado en la interfaz de la herramienta para proceder a su anotación.

- **Proceso de Anotación:**

El proceso de anotación entra en juego una vez que el texto de los documentos importados ha sido cargado en la interfaz.

Esta etapa representa el proceso de anotación manual por parte del usuario de la herramienta, en el cual se gestionan las categorías de entidad y las anotaciones realizadas con ellas.

Este proceso se va retroalimentando con el de Reconocimiento de Entidades Nombradas, que provee la característica de asistencia a la anotación, agilizando con ello el trabajo a realizar por el usuario.

- **Reconocimiento de Entidades Nombradas:**

Este proceso se ejecuta cuando el usuario activa la asistencia a la anotación. Dependiendo del modo de anotación que se escoja, la asistencia se hará mediante el uso de patrones de coincidencia, el entrenamiento de un modelo de aprendizaje automático junto con los patrones de coincidencia o las predicciones del modelo de aprendizaje automático importado en el proyecto.

- **Salida**

Representa los artefactos de salida que exporta la herramienta a petición del usuario.

Entre estos artefactos destacan el dataset de documentos anotados en el proyecto y el modelo de NER entrenado por el usuario como parte de la asistencia a la anotación dentro del proceso de NER. Dicho modelo de NER habrá sido entrenado con las anotaciones y categorías de entidad personalizadas por el usuario, pudiendo ser aplicado con efectividad sobre nuevos textos relacionados y, en consecuencia, agilizando su anotación.

## 7.2. Organización del Código

En este apartado se describe la estructura que sigue el código implementado en el proyecto. Se detallará la estructura de directorios del proyecto y se mostrarán fragmentos de código de ejemplo de ciertas partes del proyecto.

El proyecto ha sido realizado utilizando el framework web Django, el cual presenta una organización de directorios bien definida.

El directorio principal del proyecto Django presenta la estructura de la Figura 7.3:

Entre los directorios y ficheros más relevantes de la imagen se encuentran los siguientes:

- **settings.py**: script autogenerado para la configuración general del proyecto. En él podemos editar diferentes ajustes del proyecto como el registro de las aplicaciones, la definición

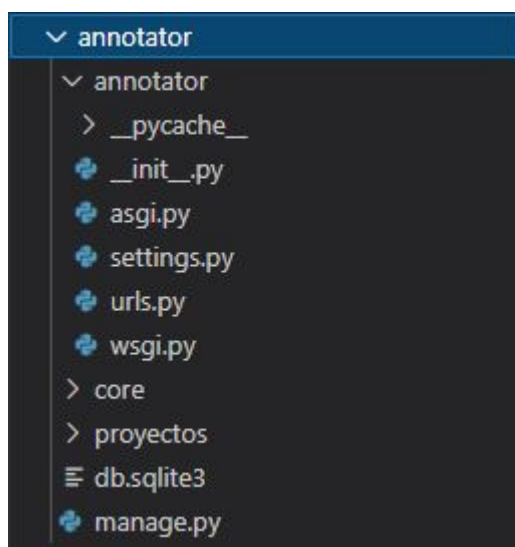


Figura 7.3: Estructura Directorio Principal del Proyecto

del directorio base donde se localizan los ficheros estáticos del proyecto o el lenguaje empleado para el reporte de advertencias y errores por parte de Django, entre otros.

- **manage.py**: script autogenerado para tareas de administración del proyecto, como crear nuevas aplicaciones o arrancar el servidor de desarrollo donde se ejecuta nuestra herramienta.
- **urls.py**: script autogenerado para la gestión de las URLs asociadas al proyecto. En él se especifica el mapeo entre las URLs y las vistas del proyecto. Para una mejor mantenibilidad del código, el mapeo URL-Vista dentro de la aplicación de la herramienta se realiza dentro de un script propio **urls.py** y se incluye una referencia al mismo dentro de este script **urls.py** global.
- **Directorio core**: representa el directorio específico de la aplicación *annotator* creada para la implementación de la herramienta.

Su estructura de directorios interna es la que aparece en la Figura 7.4:

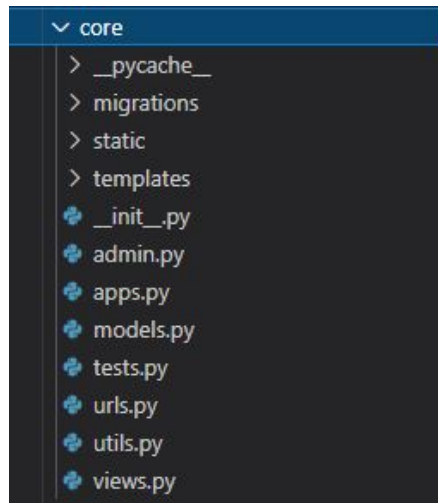


Figura 7.4: Estructura Directorio Core del Proyecto

Las partes más significativas dentro de este directorio de aplicación son las siguientes:

- **urls.py**: script utilizado para definir las URLs de la aplicación y establecer su mapeo con las vistas definidas en el script *views.py*.

El fragmento de código de la Figura 7.5 corresponde a la variable *urlpatterns* definida dentro de este script *urls.py*. En ella se definen las URLs de la aplicación junto con su mapeo a vistas.



Figura 7.5: Fragmento Código urls.py

- **views.py**: script utilizado para la implementación de las vistas de la aplicación. En él se incluyen tanto las vistas de acceso a las páginas de la aplicación como las vistas con la implementación de la lógica de negocio correspondiente a distintas funcionalidades de la herramienta. La ejecución de cada una de estas vistas se lleva a cabo a través del mapeo con la URL que las representa. Las llamadas a estas URL de las vistas tienen lugar cuando el usuario las introduce en el navegador o cuando se hacen

llamadas Ajax a ellas a través de Javascript. Este último mecanismo es el utilizado como medio principal de conexión entre el frontend y el backend de la aplicación.

El fragmento de código de la Figura 7.6 corresponde a la vista `deleteProyecto` definida dentro de este script `views.py`. Esta vista implementa la funcionalidad de eliminar un proyecto de anotación de la herramienta.

A screenshot of a code editor with a dark background and light-colored text. The code is a Python function definition for `deleteProyecto`. It starts with `def deleteProyecto(request):`, followed by an `if request.method == "POST":` block. Inside the `if` block, there are two lines: `nombreProyecto = request.POST.get('nombreProyecto')` and `res = eliminarProyecto(nombreProyecto)`. After the `if` block, there is a `return HttpResponse(res)` statement. The code is color-coded: `def` is blue, `deleteProyecto` is yellow, `request` is blue, `:` is white, `if` is blue, `request.method` is blue, `==` is white, `"POST"` is white, `:` is white, `nombreProyecto` is yellow, `=` is white, `request.POST.get` is blue, `('nombreProyecto')` is white, `res` is yellow, `=` is white, `eliminarProyecto` is blue, `(nombreProyecto)` is white, `return` is blue, `HttpResponse` is blue, and `(res)` is white. There are three colored circles (red, yellow, green) in the top left corner of the code editor window.

Figura 7.6: Fragmento Código `views.py`

La función `eliminarProyecto` a la que se llama en este fragmento de código es una función definida dentro del script `models.py` empleada para borrar la información del proyecto del componente de almacenamiento.

- **utils.py**: script creado para la implementación de las funciones de lógica de negocio asociadas a cada una de las vistas de la aplicación. Estas funciones son llamadas desde cada una de las vistas y sirven como mecanismo para mantener la legibilidad del código en el script `views.py`.

El fragmento de código de la Figura 7.7 corresponde a la función `train_model_con_batches` definida dentro de este script `utils.py`. Esta función es utilizada para el entrenamiento de un modelo de aprendizaje automático a partir de las anotaciones realizadas en los documentos del proyecto. Su ejecución se lleva a cabo al activar la característica de Asistencia a la Anotación con entrenamiento de modelo de aprendizaje y patrones de coincidencia.

```

def train_model_con_batches(TRAIN_DATA, nlp, iteraciones=20, batch_size=10, dropout=0.2):
    with nlp.disable_pipes(['tok2vec', 'morphologizer', 'parser', 'lemmatizer', 'senter']):
        perdidas = []
        for i in range(iteraciones):
            random.shuffle(TRAIN_DATA)
            losses = {}

            batches = minibatch(TRAIN_DATA, size=batch_size)
            for batch in batches:
                for documento, entidades in batch:
                    doc = nlp.make_doc(documento)
                    example = Example.from_dict(doc, {"entities": entidades['entities']})
                    #Actualizar el modelo según versión Spacy 3.0 o superior
                    nlp.update([example], losses=losses, drop=dropout)

            print("Iteración {} -> \t Valor de pérdida: {}".format(i, losses))
            perdidas.append(losses)

    return (nlp, perdidas)

```

Figura 7.7: Fragmento Código utils.py

- **models.py:** script creado para la implementación de las funciones que dan acceso al componente de almacenamiento de la herramienta. Dichas funciones son utilizadas para recuperar y publicar datos en el componente de almacenamiento, sirviendo como medio para mantener un acceso aislado a la parte de almacenamiento de los proyectos de anotación.

El fragmento de código de la Figura 7.8 corresponde a la función `eliminarProyecto` definida dentro de este script `models.py`. Esta función elimina toda la información asociada al proyecto cuyo nombre se pasa como parámetro.

- **Directorio templates:** directorio que contiene las plantillas HTML asociadas a cada una de las páginas de la herramienta.

Estas páginas son: `inicio.html`, `herramienta.html` y `expresiones.html`.

- **Directorio static:** directorio empleado para almacenar los recursos estáticos de la aplicación, tales como hojas de estilo, ficheros javascript, imágenes, etc.

Dentro de este directorio almacenamos los ficheros CSS (`inicio.css`, `herramienta.css` y `expresiones.css`), los ficheros javascript (`inicio.js`, `herramienta.js` y `expresiones.js`) y las imágenes e iconos incluidos en la herramienta.

A continuación, se presentan otros 2 fragmentos de código.

El fragmento de código de la Figura 7.9 se corresponde con reglas de estilo aplicadas para dar formato a la barra de navegación superior de la herramienta.

El fragmento de código de la Figura 7.10 representa una llamada Ajax realizada para obtener los proyectos de anotación existentes actualmente en la herramienta.

```
def eliminarProyecto(nombreProyecto):
    # Eliminar proyecto del fichero proyectos.json
    jsonProyectos = obtenerJsonProyectos()

    for numero, proyecto in enumerate(jsonProyectos['proyectos']):
        if proyecto['nombre'] == nombreProyecto:
            jsonProyectos['proyectos'].pop(numero)

    # Escribir el objeto JSON modificado
    pathJson = os.path.join(os.getcwd(), "proyectos", "proyectos.json")
    with open(pathJson, 'w') as json_file:
        json.dump(jsonProyectos, json_file)

    # Eliminar directorio asociado al proyecto actual
    pathDirectorioProyecto = os.path.join(os.getcwd(), "proyectos", nombreProyecto)
    shutil.rmtree(pathDirectorioProyecto)

    return(True)
```

Figura 7.8: Fragmento Código models.py

```
.navbar {
    overflow: hidden;
    background-color: #333;
    height: 40px;
}

.navbar a, .navbar label {
    float: left;
    font-size: 17px;
    color: white;
    text-align: center;
    padding: 10px 12px;
    text-decoration: none;
}
```

Figura 7.9: Fragmento Código herramienta.css

```
function llamadaProyectosExistentes() {
  return new Promise((resolve, reject) => {
    $.ajax({
      url: '../getJsonProyectos/',
      type: 'GET',
      dataType: 'json',
      success: function (data) {
        resolve(data);
      },
      error: function (error) {
        reject(error);
      },
    });
  });
}
```

Figura 7.10: Fragmento Código herramienta.js

- **Directorio proyectos:** este directorio incluye la configuración y datos asociados a cada proyecto de anotación gestionado en la herramienta. Por lo tanto, constituye el componente principal de almacenamiento de la herramienta.

Este directorio *proyectos* presenta la estructura de la Figura 7.11:

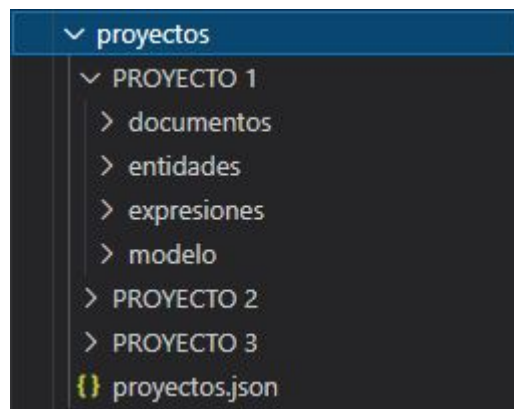


Figura 7.11: Estructura Directorio Proyectos del Proyecto

En él se encuentra el fichero de configuración general de los proyectos (*proyectos.json*) y cada uno de los directorios asociados a los proyectos de anotación individuales. El fichero *proyectos.json* contiene los nombres de todos los proyectos actuales en la aplicación junto con sus parámetros de importación y entrenamiento establecidos por el usuario.

Como ya sabemos, los proyectos de anotación actúan como contenedor para el resto de elementos del dominio de la herramienta. Dentro de la configuración de cada proyecto mantendremos los siguientes elementos:

- **documentos:** representan los textos importados por el usuario a la aplicación, listos para ser cargados en la interfaz de usuario. El texto extraído de los ficheros PDF importados por el usuario a la herramienta se almacena dentro de este directorio, no siendo necesario aplicar la etapa de preprocesamiento sobre los PDF importados cada vez que se accede al proyecto de anotación.

Igualmente, en este directorio se almacena el dataset de documentos anotados en el proyecto, posibilitando su reconstrucción cada vez que se accede de nuevo al proyecto de anotación.

- **entidades:** este directorio contiene un fichero JSON con las definiciones de las categorías de entidad creadas en el proyecto. En él se mantiene la información relativa al nombre y color de cada categoría de entidad.
- **expresiones:** este directorio contiene un fichero JSON con el nombre y definición de las expresiones regulares creadas en el proyecto. Así mismo, incluye la asociación entre las expresiones regulares y las categorías de entidad a las que representan, liberando al usuario de tener que establecer dichas asignaciones entre expresiones y categorías cada vez que abre el proyecto de anotación en la herramienta.
- **modelo:** en este directorio se almacena tanto el modelo de NER entrenado a lo largo del proyecto a petición del usuario, al activar la correspondiente opción de asistencia a la anotación, como un posible modelo preentrenado importado por el usuario a la herramienta. Este último modelo de aprendizaje importado al proyecto será recuperado por una función de la capa de acceso a datos (dentro del script *models.py*) cada vez que se active el modo de asistencia a la anotación correspondiente.

### 7.3. Herramientas Empleadas

En esta sección se listan las herramientas y utilidades empleadas a lo largo del desarrollo del proyecto, ofreciendo una breve descripción y la motivación de su uso en el proyecto.

Para una mejor organización, se presentan dichas herramientas agrupadas en distintos grupos según la categoría a la que pertenecen.

- **Entorno de Desarrollo**

- **VirtualBox:** software de virtualización de escritorio que nos permite crear máquinas virtuales con un sistema operativo distinto del instalado en el equipo anfitrión. El desarrollo de este proyecto se ha llevado a cabo en una máquina virtual con sistema operativo Ubuntu Desktop 20.04, una distribución de Linux.



- **Visual Studio Code:** entorno de desarrollo integrado (IDE) para el programación de proyectos con soporte para multitud de lenguajes de programación gracias al uso de extensiones instalables.

En nuestro caso, se utiliza como editor de código fuente para el desarrollo de la herramienta y se instalaron varias extensiones para mejorar su soporte al trabajar con Python y los lenguajes de programación web como HTML, Javascript y CSS.

- **Anaconda:** distribución open source de Python empleada ampliamente en los campos de ciencia de datos y aprendizaje automático. Cuenta con herramientas integradas para distintas finalidades y facilita la instalación directa de paquetes de funcionalidades y gestión de entornos de desarrollo virtuales.

En este proyecto se ha utilizado la aplicación Jupyter Notebook, integrada en la distribución Anaconda, para programar ejemplos y pruebas antes de codificar ciertas funcionalidades en la aplicación web.

Así mismo, se ha trabajado dentro un entorno virtual con la versión 3.9 de Python y en el que se han instalado los paquetes y dependencias necesarias. Esta es una buena práctica en Python para mantener un entorno de desarrollo aislado que permita instalar paquetes de diferentes versiones sin interferir con otras versiones instaladas en el entorno principal.

#### ■ Tecnologías Web

- **Django:** framework para el desarrollo web en Python que sigue el patrón Modelo-Vista-Plantilla, una variante del popular patrón de diseño Modelo-Vista-Controlador. En nuestro caso, Django se utiliza como framework para el desarrollo de la aplicación web.

- **HTML:** lenguaje de etiquetas utilizado para diseñar, estructurar y desplegar páginas web y su contenido. Se emplea para el desarrollo de las interfaces de usuario en un sitio web.

Se utiliza en el desarrollo del frontend de la aplicación web.

- **CSS:** lenguaje de estilos utilizado para describir la presentación visual de la interfaz de usuario de páginas web. Utilizado principalmente para diseñar y dar estilo a las páginas web.

Se utiliza en el desarrollo del frontend de la aplicación web.

- **Javascript:** lenguaje de programación interpretado utilizado principalmente en el lado cliente y dirigido a la creación de sitios web dinámicos.

Se utiliza en el desarrollo del frontend de la aplicación web.

- **jQuery:** biblioteca multiplataforma de Javascript utilizada para simplificar la interacción con los documentos HTML, la manipulación del árbol DOM, las llamadas asíncronas mediante Ajax o el manejo de eventos, entre otras.

Concretamente, en este proyecto se ha hecho uso de JQuery para simplificar el código asociado a las llamadas Ajax a vistas de la aplicación para la conexión del frontend con el backend de la aplicación.

- **JSON:** formato de texto ligero para el intercambio de datos estructurados independiente de la tecnología involucrada. Surge como alternativa a XML por su facilidad de lectura y escritura.

Dentro de nuestra herramienta, la notación JSON es utilizada en el fichero de configuración de proyectos, en la exportación de los ficheros de expresiones regulares y en el intercambio de datos entre el frontend y el backend de la aplicación.

- **AJAX:** técnica de desarrollo web para la implementación de aplicaciones web asíncronas en las que poder realizar peticiones HTTP desde el cliente al servidor sin necesidad de recargar la página web, comúnmente a través de Javascript.

En nuestro proyecto, es de gran utilidad para la conexión entre el frontend y el backend de la aplicación web.

### ■ Librerías de Python

Python es un lenguaje de programación interpretado y multiparadigma que hace un especial énfasis en la legibilidad de su código. Dispone de multitud de librerías open source que pueden ser instaladas e importadas fácilmente.

Python es el lenguaje utilizado para el desarrollo del backend de nuestra aplicación.

Las principales librerías de Python utilizadas son las siguientes:

- **Spacy:** librería open source especializada en multitud de tareas de Procesamiento del Lenguaje Natural sobre grandes volúmenes de texto. Proporciona modelos pre-entrenados para resolver tareas de Procesamiento del Lenguaje Natural de una forma directa.

En este proyecto, Spacy es utilizado tanto para la segmentación en sentencias de los documentos como para la tarea de NER, gracias a sus componentes de pipeline *senter* y *ner*, respectivamente.

- **Pytesseract:** librería open source que actúa como wrapper (envoltorio) encapsulando las funcionalidades ofrecidas por el motor OCR Tesseract, permitiendo invocarlas dentro de código Python.

Dentro de nuestro proyecto, se utiliza en el proceso de Reconocimiento Óptico de Caracteres para la obtención del texto contenido en los ficheros PDF importados por el usuario a la herramienta.

- **pdf2image:** librería open source para la conversión de las páginas de un fichero PDF en imágenes.

Se utiliza dentro de la etapa de preprocesamiento para obtener las imágenes asociadas a cada una de las páginas de los ficheros PDF importados por el usuario a la herramienta.

- **Pillow:** librería open source, derivada de la librería primigenia PIL, empleada para la manipulación y tratamiento de imágenes.

Se utiliza dentro de la etapa de preprocesamiento para abrir las imágenes asociadas a las páginas de los ficheros PDF y recortarlas.

### ■ Modelado de Diagramas

- **StarUML**: herramienta de pago con versión gratuita para la creación de diagramas UML. Contiene elementos seleccionables predefinidos relacionados con múltiples tipos de diagrama.

Se ha utilizado para construir el *Diagrama de Casos de Uso*, el *Diagrama de Clases* y los *Diagramas de Secuencia*.

- **Draw.io**: herramienta web gratuita para el diseño de diagramas de software. Consta de una gran variedad de figuras y elementos editables a medida por el usuario para componer sus diagramas.

Se ha utilizado para elaborar el *Árbol de Características*, el *Diagrama Entidad-Relación*, los *Bocetos de Interfaz de Usuario*, el *Pipeline de Procesamiento* de la Herramienta y los *Diagramas de Arquitectura*.

- **Tom's Planner**: herramienta web de pago con versión gratuita destinada a la planificación y gestión de proyectos.

Se ha utilizado para construir los *Diagramas de Gantt* de la Planificación Temporal del proyecto.

- **Creately**: herramienta web de pago con versión gratuita para el diseño de diagramas de todo tipo.

Se ha utilizado para la creación del *Modelo Lógico de Datos* de la herramienta.

### ■ Redacción de Memoria

- **TeXstudio**: editor open source de **LaTeX**, un sistema de escritura de textos orientado a la creación de documentos de alta calidad. Al contrario que los editores de texto de tipo WYSIWYG (What You See Is What You Get), la filosofía de LaTeX reside en separar el contenido a redactar del formato, delegando esto último al procesador de LaTeX.

La presente memoria está escrita con LaTeX.

- **Carbon**: herramienta web gratuita para la creación de imágenes personalizadas de fragmentos (snippets) de código.

Los fragmentos de código fuente que aparecen en esta memoria han sido formateados con Carbon.

## 7.4. Tecnologías Involucradas

En este apartado se explican los detalles de cada una de las soluciones concretas elegidas para implementar las tecnologías involucradas dentro del desarrollo del proyecto.

### 7.4.1. Anotación

Las anotaciones realizadas por el usuario de nuestra herramienta dentro de los documentos importados a un proyecto de anotación deben ser almacenadas en algún formato para su posterior carga o exportación de cara al usuario.

En este proyecto, la solución propuesta para almacenar los documentos anotados pasa por utilizar el formato básico de anotación IOB, más concretamente la versión IOB2. Este formato de anotación consiste en la asignación de una etiqueta a cada uno los tokens (palabras, signos de puntuación o espacios) de un texto para indicar una de las siguientes condiciones:

- **Etiqueta I - Inside:** El token es parte de una entidad nombrada y no es el primero.
- **Etiqueta O - Outside:** El token no pertenece a ninguna entidad nombrada.
- **Etiqueta B - Beginning:** El token es el primero de una entidad nombrada.

*Nota:* además, para los tokens del texto que corresponden a entidades nombradas o partes de ellas, también se especificará el nombre de la categoría de entidad asociada.

Alternativamente, se provee al usuario de otra opción de exportación del dataset de documentos anotados en un formato binario de Spacy. Este formato de exportación supone una restricción de implementación, pues no es un estándar de facto sino un formato propietario de Spacy, pero aún así es totalmente válido y puede ser importado a nuestra herramienta para su carga en un proyecto de anotación.

### 7.4.2. Reconocimiento de Entidades Nombradas

En este proyecto, la tarea de NER entra en juego cuando el usuario activa el proceso de asistencia a la anotación desde la herramienta. A la hora de implantar el componente de NER que realice las predicciones de entidades nombradas sobre los documentos de un proyecto de anotación, se plantean 2 propuestas de solución acordes al modo de asistencia a la anotación que active el usuario.

La primera solución se basa en el uso de patrones de coincidencia definidos a partir de las asignaciones entre expresiones regulares y categorías de entidad definidas por el usuario dentro del proyecto de anotación. Estos patrones de coincidencia son aplicados sobre los documentos del proyecto para obtener las entidades nombradas que se encuentran dentro de ellos. En nuestro caso de uso, con documentos digitalizados, esta primera propuesta presenta una clara deficiencia al estar muy condicionados los patrones de coincidencia a que aparezcan las entidades nombradas tal cual se espera. A este respecto, la eficacia de los patrones de coincidencia se vería altamente influenciada por la bondad de la salida devuelta por el motor OCR de digitalización de documentos.

La segunda propuesta ofrecida por nuestra herramienta se basa en la construcción de un pipeline de procesamiento que combina el entrenamiento de un modelo de NER a partir de las anotaciones existentes en los documentos del proyecto con el uso de los patrones de coincidencia. Esta solución requiere un mayor tiempo de realización, pero consigue una mayor eficacia en las

predicciones realizadas y permite paliar algunas de las limitaciones comentadas de la primera solución.

Ambas opciones de asistencia a la anotación se implementan a través de la librería de Procesamiento del Lenguaje Natural Spacy, la cual se describe en profundidad a continuación.

## Spacy

Spacy es una librería open source de Python destinada a realizar tareas de Procesamiento del Lenguaje Natural sobre grandes volúmenes de texto.

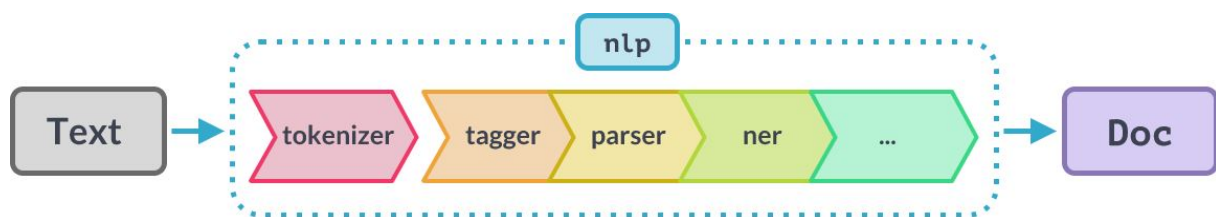


Figura 7.12: Pipeline de Procesamiento de un texto

Proporciona pipelines de procesamiento descargables con componentes preentrenados para realizar las tareas que se requieran sobre los textos pasados como entrada. Cada componente de pipeline predefinido está pensado para resolver una tarea concreta de Procesamiento del Lenguaje Natural. Según la tarea que queramos implementar, estos componentes podrán ser ordenados, añadidos o eliminados a conveniencia según se requiera.

Spacy dispone de los siguientes componentes específicos para integrar en el pipeline de procesamiento de un texto:

- **tokenizer:** este siempre es el primer componente de todo pipeline de procesamiento de textos y su funcionamiento consiste en la división en tokens del texto recibido como entrada.

Esta tarea la realiza teniendo en cuenta no solo los espacios en blanco, sino también las reglas y estructuras propias del lenguaje indicado en la creación del pipeline. Además, a partir de los tokens del texto derivados, se encarga de crear el objeto de la clase Doc que permitirá al resto de componentes del pipeline tratar con una representación manejable del texto.

- **tagger:** este componente es el encargado de asignar las etiquetas de las partes del discurso (POS) a los tokens del texto.

Dentro de las categorías gramaticales que se pueden establecer para cada token destacan los siguientes:

Este componente crea la propiedad Token.tag.

- **parser:** este componente asigna, a nivel de token, etiquetas de dependencia sintáctica entre los tokens del texto y delimita las partes principales y sentencias que componen el texto.

Etiqueta POS	Significado	Etiqueta POS	Significado
ADJ	Adjetivo	ADP	Adposición
ADV	Adverbio	AUX	Verbo Auxiliar
CONJ	Conjunción coordinante	DET	Determinante
INTJ	Interjección	NOUN	Nombre
NUM	Numeral	PART	Partícula
PRON	Pronombre	PROPN	Nombre propio
PUNCT	Signo de puntuación	SCONJ	Conjunción subordinante
SYM	Símbolo	VERB	Verbo
X	Otro		

Tabla 7.1: Etiquetas de Categorías Gramaticales - Part-of-speech tags

Este componente crea las propiedades `Token.head`, `Token.dep`, `Doc.sents` y `Doc.noun_chunks`.

- **ner**: este componente es el encargado de predecir las entidades nombradas dentro de un texto, ya sea en forma de tokens unitarios o secuencias de tokens.

Los tipos de entidades nombradas soportados por Spacy son los siguientes.

No obstante, el usuario puede crear nuevos tipos de entidades nombradas según necesite.

Para la agrupación de tokens contiguos Spacy utiliza objetos de la clase `Span`, los cuales representan secuencias de 1 o más tokens del texto. Por otra parte, también define etiquetas en formato IOB para cada token.

Este componente crea las propiedades `Doc.ents`, `Token.ent_iob` y `Token.ent_type`.

- **lemmatizer**: este componente establece para cada token del texto el lema del que deriva. El lema es la base de la que deriva una palabra flexiva. Para determinar estos lemas se hace uso de la información definida en las etiquetas POS creadas por el componente tagger.

Este componente crea la propiedad `Token.lemma`.

- **textcat**: este componente se encarga de clasificar un texto completo en una o más categorías. Para ello, analiza su contenido y hace una predicción de la categoría o categorías más probables. Desde la versión 3.0 de Spacy este componente ha sido dividido en 2 distintos, según se quiera asignar al documento una única categoría (**textcat**) o varias categorías (**textcat\_multilabel**).

Este componente crea la propiedad `Doc.cats`.

Todos estos componentes llevan a cabo la asignación de etiquetas al documento o a sus tokens por medio de la creación de nuevas propiedades en ellos. Una vez creado el objeto `Doc`

Tipo Entidad	Descripción	Tipo Entidad	Descripción
PERSON	Nombres de personas y personajes ficticios	NORP	Nacionalidades, comunidades religiosas y grupos políticos
FAC	Nombres de construcciones como edificios, aeropuertos y puentes	ORG	Nombres de organizaciones, instituciones y compañías
GPE	Localizaciones geográficas concretas como ciudades, estados y países	LOC	Localizaciones geográficas generales como cordilleras, océanos y ríos
PRODUCT	Objetos tangibles de todo tipo	EVENT	Sucesos relevantes como guerras, batallas y eventos deportivos
WORK_OF_ART	Títulos de obras de arte, canciones y novelas	LAW	Nombres de documentos o casos legislativos
LANGUAGE	Idiomas	DATE	Fechas y períodos de tiempo
TIME	Expresiones de tiempo menores que un día	PERCENT	Porcentajes tanto en formato alfabético como numérico
MONEY	Cantidades monetarias tanto en formato alfabético como numérico	QUANTITY	Cantidades de medición como pesos, alturas o distancias
ORDINAL	Número ordinal en formato numérico o alfabético	CARDINAL	Número cardinal en formato numérico o alfabético

Tabla 7.2: Tipos de Entidades Nombradas - Named Entity types

por parte del componente tokenizer, la operativa general del resto de componentes del pipeline será tomar como entrada el objeto Doc, procesar dicho objeto y devolver como salida hacia el siguiente componente del pipeline el objeto Doc refinado. Además, aparte de estos componentes preestablecidos el desarrollador puede crear un componente de pipeline a medida (custom) para integrar en un pipeline ya definido.

Por tanto, Spacy permite crear pipelines de procesamiento desde cero, sin ningún componente preentrenado, o ya con una serie de componentes preentrenados. Así mismo, un pipeline ya definido podrá ser editado por el desarrollador según sus necesidades. Para el primer propósito se utiliza el método **spacy.blank** que recibe como parámetro un código de lenguaje concreto según el idioma de los textos que se desean procesar con el pipeline. En el segundo caso se utiliza el método **spacy.load** que recibe como parámetro el nombre de un paquete con una configuración preestablecida para formar el pipeline de procesamiento. Dicho paquete debe haber sido descargado previamente mediante el comando **python -m spacy download <nombre\_paquete>**.

Ambos métodos devuelven un objeto de la clase `Language` que representa el pipeline de procesamiento de texto, el cual podremos consultar, modificar, entrenar o evaluar posteriormente con nuestros propios datos acorde a nuestras necesidades. Algunas de las modificaciones que se pueden realizar son la inclusión de nuevos componentes de pipeline para realizar nuevas tareas sobre los textos procesados, cambiar su orden para aplicar unos componentes antes que otros durante el procesamiento de los textos...

Como se ha visto, `Spacy` es una librería muy completa para integrar dentro de un proyecto en Python que permite realizar multitud de tareas relacionadas con el PLN, y no solo la tarea de NER involucrada en nuestro proyecto.

### 7.4.3. Reconocimiento Óptico de Caracteres

En nuestro proyecto, la tecnología OCR se emplea dentro del proceso de obtención del texto contenido en los ficheros PDF que el usuario puede importar a la herramienta para su posterior anotación.

Entre los distintos motores OCR existentes para realizar el proceso de digitalización de documentos, se ha optado por `Tesseract` debido a su buen rendimiento y facilidad de uso dentro de un entorno Python a través de su wrapper `Pytesseract`.

A continuación, se describe en detalle la operativa y funcionalidades ofrecidas por `Tesseract`.

#### **Tesseract OCR**

`Tesseract` es un motor de Reconocimiento Óptico de Caracteres mantenido por Google y liberado bajo licencia Apache para su uso por parte de la comunidad.

Actualmente, `Tesseract` es considerado como uno de los mejores motores OCR y la posibilidad de ser integrado dentro de Python lo hace uno de los más utilizados. Para su integración en Python existe el mencionado `Pytesseract`, un wrapper de código abierto que encapsula la funcionalidad principal de `Tesseract` y permite invocarla a través de funciones definidas en el lenguaje Python.

Por otro lado, `Tesseract` tiene soporte para multitud de idiomas diferentes gracias a la disponibilidad de datasets con datos específicos de cada lenguaje. Por defecto utiliza el idioma inglés, pero a través de la línea de comandos o bien en el código Python puede proporcionarse el modelo de lenguaje concreto del idioma que se necesite para tratar con la imagen que contiene el texto a extraer.

Además, su última versión ha añadido, como parte de su motor de procesamiento, el uso de redes recurrentes LSTM que pueden ser entrenadas con extensos datasets de datos para contribuir a mejorar la predicción del texto realizada.

El uso de `Tesseract` en este proyecto se hará a través de `Pytesseract`, por lo que describiremos su funcionalidad a partir de los métodos de esta librería.

Las principales funciones de la librería `Pytesseract` son las siguientes:

- **`image_to_string`**: devuelve la representación en formato string del texto derivado de la image recibida como entrada.



- **image\_to\_boxes**: devuelve los caracteres reconocidos en la imagen e información acerca de los límites de las áreas dentro de la imagen (bounding boxes) que los contienen.
- **image\_to\_data**: devuelve una versión enriquecida de la información aportada por el método anterior (**image\_to\_boxes**), incluyendo valores de confianza.
- **image\_to\_osd**: proporciona diversa información como la orientación del texto en la imagen.

Todas estas funciones descritas soportan el paso de ciertos parámetros opcionales para diferentes propósitos. Algunos de ellos son los siguientes:

- **image**: representa un objeto con la imagen preprocesada por la librería PIL, array de la librería Numpy con imágenes o la ruta de la imagen en el sistema de ficheros. Este parámetro establece la imagen a ser procesada por el motor OCR.
- **lang**: indica el idioma del modelo con el que procesar el texto de la imagen. Por defecto su valor es **eng**, correspondiente al idioma inglés.
- **output\_type**: representa el tipo de salida que devolverá la función. Por defecto siempre se devuelve un string.
- **config**: esta opción permite al desarrollador pasar parámetros personalizados adicionales para ser aplicados en el procesamiento.

Por último, se considera oportuno profundizar en el funcionamiento que subyace a las redes recurrentes LSTM utilizadas por Tesseract OCR, ya que éstas tienen un papel fundamental dentro del proceso.

### Redes Recurrentes LSTM

Las redes LSTM (Long Short-Term Memory) son un tipo especial de red neuronal recurrente capaces de aprender dependencias a largo plazo. Su utilización en tareas de Procesamiento del Lenguaje Natural es especialmente útil al permitir recordar estructuras, patrones o palabras ocurridas anteriormente. Así, una red LSTM podrá inferir la información correcta en base a lo que ya ha ocurrido.

El motor Tesseract OCR incluyó las redes LSTM a partir de su versión 4.0, tras lo cual ha mejorado su rendimiento pudiendo predecir partes del texto relacionándolas con otras que han sido procesadas  $x$  instantes de tiempo antes. Su principal ventaja respecto de las redes recurrentes tradicionales es su capacidad para recordar relaciones a lo largo plazo, no restringiendo su aprendizaje a los instantes de tiempo más cercanos. Por contra, las redes recurrentes tradicionales adolecen de un fenómeno conocido como *Vanishing Gradient*, por el cual una parte del texto que ha sido procesada se olvida progresivamente a medida que se aleja en el tiempo (según se van aprendiendo partes más recientes). El funcionamiento y estructura de una red LSTM para evitar este problema se explica a continuación.

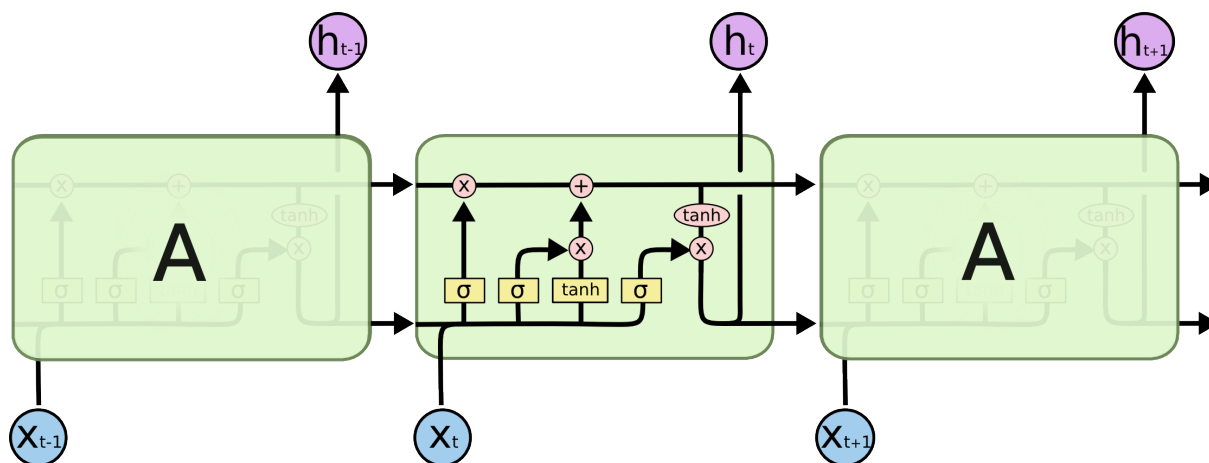


Figura 7.13: Estructura red LSTM

Una red recurrente tradicional puede verse como una cadena de módulos de red neuronal repetidos, en los que la entrada de cada módulo proviene de la salida del módulo anterior. En este caso, dicho módulo repetido tan solo se compone de una única capa con una función de activación. En cambio, en una red de tipo LSTM los módulos tienen una estructura más compleja formada por 4 capas, lo cual la permite aprender selectivamente patrones a largo plazo.

Dicha estructura aparece representada en la Figura 7.13.

Un concepto clave en las redes LSTM es el estado de la celda (cell state) que almacena los patrones aprendidos y se pasa entre los sucesivos módulos que componen la red. La información almacenada como estado de la celda se puede actualizar de forma regulada a través del uso de puertas (gates). Estas puertas están formadas por una operación matemática concreta junto con una capa de red neuronal con función de activación sigmoide que filtra la información que puede pasar al estado de la celda. La función de activación sigmoide devuelve una salida numérica entre 0 y 1, representando la probabilidad de llevar a cabo o no una determinada acción.

El funcionamiento de las 4 capas que forman un módulo de la red es el siguiente:

1. En primer lugar, se pasa por la capa 1 con función de activación sigmoide que decide la información del estado de la celda que debe desecharse.
2. Después, las capas 2 y 3 con funciones de activación sigmoide y tangente hiperbólica deciden qué información añadir al estado.
  - 1) Primero, la capa 2 decide qué valores del estado actualizar.
  - 2) Tras ello, la capa 3 crea un vector con valores candidatos para ser añadidos al estado.
  - 3) Finalmente, ambas capas se combinan para crear una actualización del estado.
3. Por último, la capa 4 con función de activación sigmoide decide la información del estado que se pasará como salida del módulo, y por consiguiente, llegará como entrada al siguiente módulo. El estado de la celda que llegará al siguiente módulo de la red será una versión

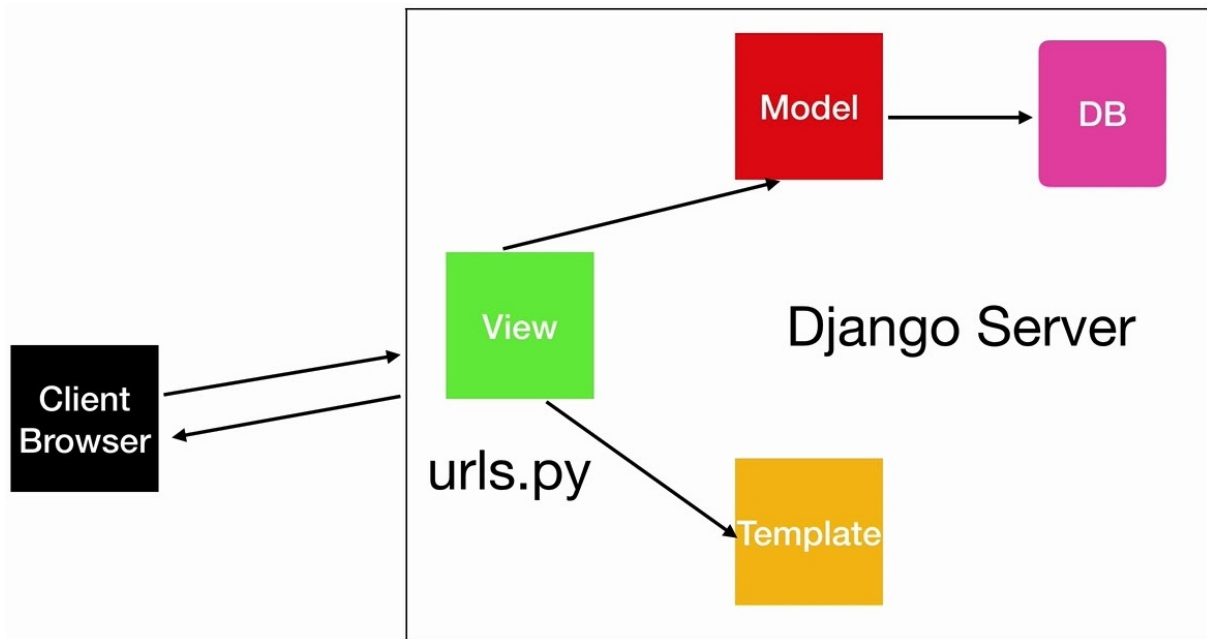


Figura 7.14: Patrón MVT de Django

filtrada del estado actual, resultado de combinar la capa 4 con una función de activación tangente hiperbólica aplicada sobre el estado de la celda existente.

La lógica matemática subyacente al funcionamiento de este tipo de redes es más compleja y se omite en esta explicación. No obstante, se incluye la referencia a un artículo donde se explican estas consideraciones en mayor profundidad.

#### 7.4.4. Aplicación Web

Para la implementación de la aplicación web de nuestra herramienta se ha optado usar el framework web Django, en detrimento de otras opciones consideradas como Flask.

A continuación, se describen los conceptos básicos de Django, la arquitectura que soporta y las ventajas que proporciona que motivan su uso.

Concretamente, Django es un framework orientado al desarrollo web dentro de un entorno Python basado en una adaptación propia del conocido patrón de diseño Modelo-Vista-Controlador (MVC), el patrón Modelo-Vista-Template (MVT).

La estructura básica de este patrón de diseño se basa en los 3 siguientes componentes principales, representados en la Figura 7.14.

- **Modelo:** contiene los datos dinámicos que maneja la aplicación. Devuelve los datos correspondientes a la vista que los solicita.
- **Vista:** se mapea (corresponde) con una URL accedida desde el navegador y solicita los datos requeridos al modelo para entregárselos a una plantilla.

- **Template** (Plantilla): representa un fichero con datos que se devuelve al navegador del cliente como respuesta a la petición realizada.

La motivación de utilizar Django antes que otras alternativas existentes, como Flask, radica en algunas de las ventajas que ofrece. Entre dichas ventajas destacan las siguientes.

- Uso de una versión del patrón Modelo-Vista-Controlador, habilitando el desarrollo ágil y la reusabilidad.
- Buen rendimiento y flexibilidad, enfocado a la escalabilidad.
- Utilidad de administración para facilitar al desarrollador la gestión de la aplicación web.
- Amplia documentación y comunidad de desarrolladores en Internet.

El uso de Django como framework web nos permite el desarrollo rápido de nuestra aplicación web y su integración con los modelos de NER que entrenaremos en lenguaje Python.

# Capítulo 8

## Pruebas

Las pruebas son uno de los mecanismos fundamentales para la evaluación de un sistema. Constituyen una forma de probar la corrección y eficacia de las funcionalidades ofrecidas por el sistema. Realizar pruebas sobre un sistema software resulta de gran utilidad no solo para verificar el funcionamiento esperado en el sistema sino también para detectar posibles fallos de implementación o concepto.

Principalmente, existen 2 tipos de pruebas de software: las **pruebas de caja blanca** y las **pruebas de caja negra**.

Las pruebas de caja blanca se centran en el estudio de la estructura interna del código y en los detalles concretos de su implementación, es decir, analizan el flujo de proceso de la funcionalidad para determinar por qué partes del código se va pasando en cada momento. Estas pruebas de caja blanca se han ido realizando durante la implementación de las distintas funcionalidades de la herramienta, estudiándose el flujo de proceso asociado a diferentes entradas que se iban aportando a la funcionalidad. Especialmente, han sido tratadas para probar las funcionalidades relacionadas con la gestión de las anotaciones en los textos.

Por otra parte, en este proyecto se han ejecutado pruebas de caja negra sobre las principales funciones de nuestra herramienta. Las pruebas de caja negra se utilizan para verificar las funcionalidades del sistema sin tener en cuenta la estructura interna y detalles del código asociado a una funcionalidad, de ahí su nombre. Este tipo de pruebas, tal como se muestra en la Figura 8.1, consisten en especificar unas entradas concretas de un escenario de uso y comprobar si el resultado de salida coincide con el esperado.



Figura 8.1: Esquema Pruebas de Caja Negra

A continuación, se presentan varias pruebas de caja negra ejecutadas sobre nuestra herramienta. Se incluyen las pruebas asociadas a los escenarios de uso más reseñables, obviando el resto por simplicidad o similitud con las mostradas.

<b>PCN-01</b>	<b>Renombrar un proyecto de anotación existente</b>
Propósito	Probar que se impide renombrar un proyecto con el nombre de otro proyecto existente.
Prerrequisitos	Proyecto de anotación abierto en la herramienta.
Datos de entrada	Un nombre que ya pertenece a otro proyecto existente.
Resultado esperado	El sistema impide renombrar el proyecto e informa al usuario de la existencia de otro proyecto con el nombre introducido.
Resultado obtenido	Correcto.

Tabla 8.1: Prueba de Caja Negra PCN-01: Renombrar un proyecto de anotación existente.

<b>PCN-02</b>	<b>Crear una nueva expresión regular</b>
Propósito	Probar que se impide crear una nueva expresión regular con la misma definición que otra expresión existente.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Alguna expresión regular existente en el proyecto.
Datos de entrada	Un nombre válido cualquiera y una definición repetida para la nueva expresión regular.
Resultado esperado	El sistema impide la creación de la expresión regular e informa al usuario de que la definición introducida ya pertenece a otra expresión regular.
Resultado obtenido	Correcto.

Tabla 8.2: Prueba de Caja Negra PCN-02: Crear una nueva expresión regular.

<b>PCN-03</b>	<b>Añadir una nueva anotación en un texto</b>
Propósito	Probar que se impide crear una nueva anotación en caso de solaparse con otra existente.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Algún documento importado en el proyecto de anotación. Alguna anotación realiza en los documentos del proyecto.
Datos de entrada	Una selección en el texto con límites de inicio y fin definidos.
Resultado esperado	El sistema impide la creación de la nueva anotación en el texto e informa al usuario de que la nueva anotación no puede solaparse con otra existente.
Resultado obtenido	Correcto.

Tabla 8.3: Prueba de Caja Negra PCN-03: Añadir una nueva anotación en un texto.

<b>PCN-04</b>	<b>Editar los límites de una anotación existente</b>
Propósito	Probar que se impide modificar los límites de una anotación en caso de ser estos inválidos.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Algún documento importado en el proyecto de anotación. Alguna anotación realizada en los documentos del proyecto.
Datos de entrada	Nuevos límites de la anotación inválidos. Por ejemplo: el límite de fin de la anotación se encuentra en mitad de una palabra.
Resultado esperado	El sistema impide la modificación de los límites de la anotación y retorna sus valores a los establecidos anteriormente.
Resultado obtenido	Correcto.

Tabla 8.4: Prueba de Caja Negra PCN-04: Editar los límites de una anotación existente.

<b>PCN-05</b>	<b>Editar el color de una categoría de entidad</b>
Propósito	Probar que se impide cambiar el color de una categoría de entidad en caso de que el nuevo color ya esté asignado a otra categoría del proyecto.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Alguna categoría de entidad existente en el proyecto de anotación.
Datos de entrada	Un color que ya está asignado a otra categoría de entidad.
Resultado esperado	El sistema impide la asignación del nuevo color a la categoría de entidad e informa al usuario de que ya está asignado a otra categoría de entidad.
Resultado obtenido	Correcto.

Tabla 8.5: Prueba de Caja Negra PCN-05: Editar el color de una categoría de entidad.

<b>PCN-06</b>	<b>Activar asistencia con patrones de coincidencia</b>
Propósito	Probar que se impide activar la asistencia a la anotación con patrones de coincidencia en caso de no tener ninguna expresión regular asociada a una categoría de entidad en el proyecto.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Algún documento importado en el proyecto de anotación.
Datos de entrada	Selección de la opción de Activar Asistencia con patrones de coincidencia sin tener ninguna expresión regular asociada a una categoría de entidad en el proyecto.
Resultado esperado	El sistema impide activar el modo de asistencia a la anotación e informa al usuario de la necesidad de asociar primero alguna expresión regular a una categoría de entidad.
Resultado obtenido	Correcto.

Tabla 8.6: Prueba de Caja Negra PCN-06: Activar asistencia con patrones de coincidencia.



<b>PCN-07</b>	<b>Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia</b>
Propósito	Probar que se impide activar la asistencia a la anotación con entrenamiento de modelo de aprendizaje y patrones de coincidencia en caso de no tener ninguna anotación realizada en los documentos del proyecto de anotación.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Algún documento importado en el proyecto de anotación.
Datos de entrada	Selección de la opción de Activar Asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia sin tener ninguna anotación realizada en los documentos del proyecto.
Resultado esperado	El sistema impide activar el modo de asistencia a la anotación e informa al usuario de la necesidad de realizar primero algunas anotaciones en los documentos con las que poder entrenar un modelo de aprendizaje.
Resultado obtenido	Correcto.

Tabla 8.7: Prueba de Caja Negra PCN-07: Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia.

<b>PCN-08</b>	<b>Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia</b>
Propósito	Probar que se impide activar la asistencia a la anotación con predicciones de modelo de aprendizaje importado y patrones de coincidencia en caso de no haber importado ningún modelo de aprendizaje preentrenado al proyecto.
Prerrequisitos	Proyecto de anotación abierto en la herramienta. Algún documento importado en el proyecto de anotación.
Datos de entrada	Selección de la opción de Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia sin haber importado ningún modelo de aprendizaje preentrenado al proyecto.
Resultado esperado	El sistema impide activar el modo de asistencia a la anotación e informa al usuario de la necesidad de importar primero un modelo de aprendizaje preentrenado al proyecto.
Resultado obtenido	Correcto.

Tabla 8.8: Prueba de Caja Negra PCN-08: Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia.

<b>PCN-09</b>	<b>Importar fichero PDF</b>
Propósito	Probar que se impide la capacidad de importación de fichero PDF en caso de subir el usuario a la herramienta un fichero con extensión distinta de .pdf.
Prerrequisitos	Proyecto de anotación abierto en la herramienta.
Datos de entrada	Subida de fichero con extensión distinta de .pdf en la opción de Importar PDF.
Resultado esperado	El sistema impide la capacidad de importación de fichero PDF e informa al usuario de que debe subir a la herramienta un fichero con extensión .pdf.
Resultado obtenido	Correcto.

Tabla 8.9: Prueba de Caja Negra PCN-09: Importar fichero PDF.

<b>PCN-10</b>	<b>Crear un proyecto de anotación</b>
Propósito	Probar que se impide la creación de un nuevo proyecto de anotación en caso de introducir un nombre con algún carácter inválido.
Prerrequisitos	Proyecto de anotación abierto en la herramienta.
Datos de entrada	Un nombre con algún carácter inválido para el nuevo proyecto de anotación. Por ejemplo, un nombre que incluya una barra inclinada (/).
Resultado esperado	El sistema impide la creación del nuevo proyecto de anotación e informa al usuario de los caracteres válidos que pueden componer el nombre de un proyecto.
Resultado obtenido	Correcto.

Tabla 8.10: Prueba de Caja Negra PCN-10: Crear un proyecto de anotación.

---

Igualmente, se podrían realizar pruebas para constatar que una funcionalidad funciona correctamente según lo esperado.

<b>PCN-11</b>	<b>Añadir una nueva categoría de entidad</b>
Propósito	Probar que se crea correctamente la nueva categoría de entidad en el proyecto.
Prerrequisitos	Ninguno.
Datos de entrada	Un nombre válido y un color que no esté asignado a otra categoría de entidad del proyecto.
Resultado esperado	El sistema crea correctamente la nueva categoría de entidad en el proyecto.
Resultado obtenido	Correcto.

Tabla 8.11: Prueba de Caja Negra PCN-11: Añadir una nueva categoría de entidad.



# Capítulo 9

## Manuales

En este capítulo se presentan los manuales de instalación y de uso de *Annotator*, dispuestos para la puesta en marcha de la herramienta. La existencia de estos manuales pretende facilitar tanto el aprendizaje del uso básico de la herramienta como su proceso de instalación.

### 9.1. Manual de Instalación

El presente manual de instalación tiene como finalidad mostrar los pasos necesarios para la instalación desde cero de la aplicación en un entorno de desarrollo ya sea para la continuación de su implementación o para su ejecución en local.

El desarrollo en el proyecto ha sido realizado en una máquina virtual con sistema operativo Ubuntu Desktop 20.04 creada a través del software de virtualización de escritorio VirtualBox.

El programa VirtualBox junto con su paquete de extensiones pueden ser descargados a través del siguiente enlace: <https://www.virtualbox.org/wiki/Downloads>

Así mismo, la imagen ISO correspondiente al sistema operativo Ubuntu Desktop 20.04 puede ser descargada en el siguiente enlace: <https://ubuntu.com/download/desktop>.

Una guía detallada acerca del proceso de creación de una máquina virtual en VirtualBox se omite en esta memoria. Aun así, se proporciona la siguiente referencia en la que se especifica el proceso necesario paso a paso: <https://brb.nci.nih.gov/seqtools/installUbuntu.html>

Una vez completada la instalación de la máquina virtual, se procede con los siguientes pasos de instalación requeridos, dentro ya de la máquina virtual.

En primer lugar, procederemos a instalar Anaconda en nuestra máquina virtual.

Antes de ello, es recomendable actualizar el gestor de paquetes APT (Advanced Package Tool) para actualizar sus repositorios de paquetes.

```
$ sudo apt-get update
```

Seguidamente, instalaremos *curl*, una utilidad para la transferencia de ficheros con la que poder realizar peticiones HTTP por línea de comandos.

```
$ sudo apt-get install curl
```

Ahora, ya podremos descargar la distribución de Anaconda que necesitamos.

```
$ curl -O https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh
```

También, se puede descargar la distribución Anaconda a través del siguiente enlace: <https://www.anaconda.com/products/individual#linux>

A continuación, comprobamos la autenticidad e integridad del fichero descargado mediante curl. Esta es una simple medida de seguridad preventiva.

```
$ sha256sum Anaconda3-2020.02-Linux-x86_64.sh
```

Finalmente, instalamos el fichero descargado a través del siguiente comando.

```
$ bash Anaconda3-2020.11-Linux-x86_64.sh
```

Tras completar la instalación de Anaconda, se instalará un IDE (Entorno de Desarrollo Integrado) con el que trabajar en el desarrollo del proyecto. En nuestro caso, se ha optado por trabajar con Visual Studio Code.

Éste puede ser instalado por línea de comandos.

```
$ sudo snap install --classic code
```

O, directamente, accediendo a la aplicación Ubuntu Software, preinstalada en Ubuntu Desktop 20.04.

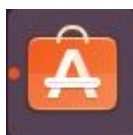


Figura 9.1: Aplicación Ubuntu Software

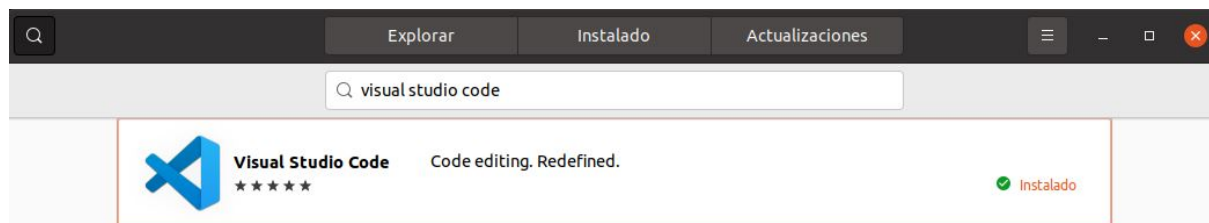


Figura 9.2: Visual Studio Code en Ubuntu Software

Una vez completada la instalación, el siguiente paso será instalar los paquetes y dependencias necesarias para el proyecto.

Primero, instalamos el motor OCR Tesseract con la siguiente instrucción.

```
$ sudo apt-get install tesseract-ocr -y
```

También, instalaremos otro paquete para los modelos de Tesseract del idioma español.

```
$ sudo apt-get install tesseract-ocr-spa -y
```

A continuación, crearemos un entorno virtual de Python, llamado **tfg**, para trabajar de una forma aislada del resto de entornos. Esta es una conocida buena práctica en Python que nos servirá para evitar conflictos entre posibles distintas versiones instaladas de un mismo paquete o librería en diferentes entornos de desarrollo.

```
$ conda create -n tfg python=3
```

Una vez creado, lo activamos.

```
$ conda activate tfg
```

Ahora, comenzaremos a instalar los paquetes junto con sus dependencias necesarias dentro del entorno de desarrollo virtual en el que nos encontramos

Primero, nos aseguramos de que la versión de pip, el instalador de paquetes de Python, esté actualizado.

```
(tfg) $ python3 -m pip install --upgrade pip
```

Empezaremos con la instalación de la librería de Procesamiento del Lenguaje Natural Spacy.

```
(tfg) $ pip install -U spacy
```

Se instalará la propia librería Spacy junto con otras dependencias asociadas.

También instalaremos un paquete de modelo preentrenado de Spacy ideal para el idioma español.

```
(tfg) $ python -m spacy download es_core_news_sm
```

Luego, este mismo paquete de modelo lo podremos cargar dentro de código Python con la siguiente instrucción, asociándolo a una variable que representará el pipeline del modelo a manipular.

```
>>> nlp = spacy.load('es_core_news_sm')
```

A continuación, instalaremos las demás librerías requeridas, las cuales son Django, Pytesseract, Pillow y pdf2image.

```
(tfg) $ pip install django
(tfg) $ pip install pytesseract
(tfg) $ python3 -m pip install --upgrade Pillow
(tfg) $ pip install pdf2image
```

Ahora, ya podemos abrir el proyecto Django en Visual Studio Code (File >Open Folder).

Nos situamos en el script *manage.py* de administración del proyecto y tras hacer click derecho seleccionar la opción *Open in Integrated Terminal*.

Esta acción nos abrirá un nuevo terminal en la ruta donde se encuentra dicho script.

Dentro del terminal abierto en Visual Studio Code introducimos el comando para activar nuestro entorno virtual.

```
$ conda activate tfg
```

A continuación, ya podemos desplegar la aplicación en el servidor de desarrollo ofrecido por Django a través del siguiente comando.

```
$ python manage.py runserver localhost:8000
```

Una vez arrancado, el servidor estará escuchando peticiones en el puerto 8000 de nuestra máquina virtual.

Finalmente, introduciendo la URL *localhost:8000/* en un navegador, comprobamos que tenemos acceso a nuestra aplicación **Annotator**.



Figura 9.3: Página de Inicio Annotator

En la entrega del proyecto se ha incluido el fichero *requirements.txt*, el cual permite instalar todos los paquetes mencionados de una sola vez.

El comando utilizado para ello sería el siguiente.

```
(tfg) $ pip install -r ./requirements.txt
```

*Nota:* el fichero *requirements.txt* con las dependencias instaladas en el entorno virtual de desarrollo ha sido generado a través del siguiente comando.



```
(tfg) $ pip freeze > requirements.txt
```

Las versiones concretas de los principales paquetes instalados se muestran en la siguiente tabla:

Paquete	Versión Instalada
Anaconda	4.10.3
Django	3.2.5
pdf2image	1.16.0
Pillow	8.3.0
pip	21.1.3
Pytesseract	0.3.8
Python	3.9.5
Spacy	3.0.6
Tesseract	4.1.1
Visual Studio Code	1.57.1

Tabla 9.1: Versiones de Paquetes Instalados

## 9.2. Manual de Usuario

El presente manual de usuario pretende servir de ayuda al usuario de la herramienta de cara a aprender los conceptos del dominio involucrados en ella, así como facilitar el aprendizaje de su uso a través de la explicación de las distintas funcionalidades que se pueden llevar a cabo con *Annotator*.

En primer lugar, se expone una breve descripción de las 3 páginas principales de la herramienta y de los mecanismos de navegación entre ellas.

### 1. **index.html:**

Esta página consta de una barra de navegación superior con las opciones de Crear Proyecto y Abrir Proyecto. Además, contiene unas definiciones de ayuda al usuario para el correcto entendimiento de los elementos del dominio de la herramienta.

Es accesible al introducir en el navegador la URL base *http://localhost:8000/* y a través de la opción *Inicio* en la barra de navegación superior de las otras 2 páginas.

### 2. **herramienta.html:**

Esta página es la que engloba la mayor parte de la funcionalidad de la herramienta, es decir, todo lo relativo a la gestión manual de anotaciones, documentos y categorías de entidad y a la capacidad de asistencia a la anotación.

Para ello, se compone de una barra de navegación superior con diferentes opciones seleccionables, una vista lateral izquierda con los resúmenes de cada una de los fragmentos del presente documento, una vista central con la sentencia actual del presente documento para su anotación manual por parte del usuario y un vista lateral derecha desplegable con las categorías de entidad existentes y las anotaciones realizadas con ellas.

Es accesible al Abrir un Proyecto en la página *index.html* y a través de la opción *Herramienta Anotación* en la barra de navegación superior de la página *expresiones.html*.

### 3. **expresiones.html:**

Esta página engloba todas las funcionalidades relativas a la gestión de las expresiones regulares de un proyecto.

Para ello, consta de una barra de navegación superior con diferentes opciones seleccionables, una vista lateral izquierda para la gestión de las expresiones regulares y su asignación a categorías de entidad y una vista lateral derecha para la visualización y gestión de las categorías de entidad existentes en el proyecto.

Es accesible a través de la opción *Gestión Regex* en la barra de navegación superior de la página *herramienta.html*.

Por otra parte, las funcionalidades ofrecidas por la herramienta han sido estructuradas en torno a 4 bloques diferentes:

1. **Gestión de Proyectos:**
2. **Gestión de Expresiones Regulares:**
3. **Anotación Manual**
4. **Asistencia a la Anotación**

A continuación, se procede a describir en detalle cada uno de ellos.

### **Gestión de Proyectos**

Dentro de la Gestión de Proyectos, el usuario puede llevar a cabo una serie de acciones sobre los proyectos de anotación.

Estas acciones referidas son las siguientes:

- **Crear Proyecto:**



Figura 9.4: Opción Crear Proyecto

Esta funcionalidad es accesible desde la página *index.html* a través de una opción en la barra de navegación superior.

La selección de esta opción abrirá un panel en la interfaz de usuario para escribir el nombre del nuevo proyecto de anotación a crear.

*Nota:* debe tenerse en cuenta que el nombre del nuevo proyecto no puede coincidir con el establecido para otro proyecto existente en la herramienta.

- **Abrir Proyecto:**



Figura 9.5: Opción Abrir Proyecto

Esta funcionalidad es accesible desde la página *index.html* a través de una opción en la barra de navegación superior.

La selección de esta opción abrirá un panel en la interfaz de usuario con un listado de los proyectos que existen actualmente en la herramienta. En éste, el usuario puede seleccionar el proyecto de anotación que desea abrir.

- **Renombrar Proyecto:**

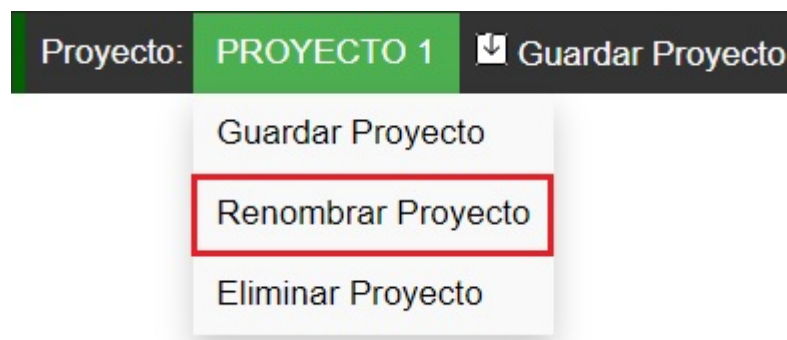


Figura 9.6: Opción Renombrar Proyecto

Esta funcionalidad es accesible desde la página *herramienta.html* a través de una opción en el menú desplegable del proyecto en la barra de navegación superior.

La selección de esta opción abrirá un panel en la interfaz de usuario para que el usuario introduzca el nuevo nombre del proyecto de anotación.

*Nota:* debe tenerse en cuenta que el nuevo nombre del proyecto no puede coincidir con el establecido para otro proyecto existente en la herramienta.

- **Eliminar Proyecto:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de una opción en el menú desplegable del proyecto en la barra de navegación superior.

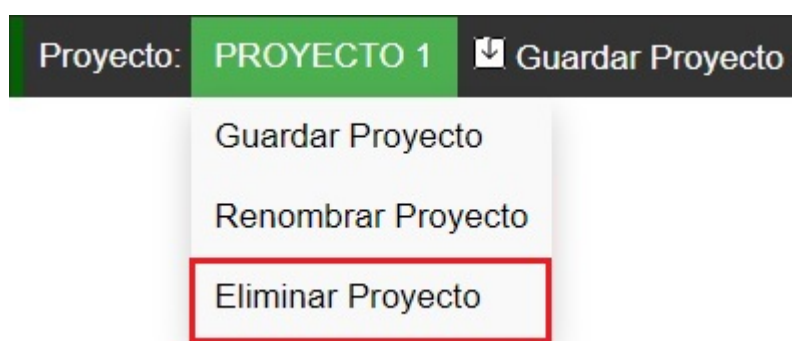


Figura 9.7: Opción Eliminar Proyecto

La selección de esta opción abrirá un panel en la interfaz de usuario para que el usuario confirme la decisión de eliminar el proyecto actual abierto en la herramienta. Alternativamente, el usuario puede cerrar el panel de decisión para descartar finalmente la eliminación del proyecto.

■ **Guardar Proyecto:**



Figura 9.8: Opción Guardar Proyecto

Esta funcionalidad es accesible desde la página *herramienta.html* a través de 2 opciones alternativas en la barra de navegación superior.

Esta opción permite al usuario guardar la configuración actual de proyecto de anotación abierto en la herramienta. Cuando esta opción es seleccionada todos los elementos pertenecientes al proyecto serán almacenados.

Entre estos elementos se incluyen los siguientes:

- Documentos importados junto con sus anotaciones asociadas
- Categorías de entidad
- Expresiones regulares junto con su asignación a categorías de entidad.

## Gestión de Expresiones Regulares

Dentro de la Gestión de Expresiones Regulares, el usuario puede llevar a cabo una serie de acciones sobre las expresiones regulares de un proyecto de anotación. El objetivo perseguido será conformar los patrones de coincidencia sobre los que procesar los documentos del proyecto para su anotación automática.

Estas acciones referidas son las siguientes:

- **Crear Expresión Regular:**

Figura 9.9: Opción Crear Expresión Regular

Esta funcionalidad es accesible desde la página *expresiones.html* a través de la opción Añadir Expresión Regular.

Esta opción permite al usuario crear una nueva expresión regular para el proyecto de anotación abierto en la herramienta. Al seleccionar la opción se muestran 2 campos de texto para introducir el nombre y la definición de la nueva expresión regular.

*Nota:* debe tenerse en cuenta que ni el nombre ni la definición de la expresión regular pueden coincidir con los de otra expresión regular existente en el proyecto.

- **Eliminar Expresión Regular:**

Figura 9.10: Opción Eliminar Expresión Regular

Esta funcionalidad es accesible desde la página *expresiones.html* a través de la opción representada por el icono de una papelera al margen de la expresión regular correspondiente.

Esta opción permite al usuario eliminar una expresión regular del proyecto de anotación actual.

*Nota:* en caso de estar asignada a alguna categoría de entidad la expresión regular a eliminar, se pedirá primero al usuario una confirmación de la eliminación de la expresión regular.

- **Editar Expresión Regular:**

Figura 9.11: Opción Editar Expresión Regular

Esta funcionalidad es accesible desde la página *expresiones.html* a través de una opción de ratón, el click derecho, sobre el nombre de una expresión regular o de forma directa para la edición de su definición.

Esta opción permite al usuario editar tanto el nombre como la definición de la expresión regular en cuestión.

*Nota:* debe tenerse en cuenta que ni el nuevo nombre ni nueva la definición de la expresión regular pueden coincidir con los de otra expresión regular existente en el proyecto.

- **Asignar Expresión Regular a Categoría de Entidad:**

## Asignación de Expresiones Regulares

	OTRA REGEX DNI	REGEX E-MAIL	REGEX DIRECCIÓN	REGEX TELÉFONO
DNI	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
E-MAIL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
DIRECCIÓN	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
TELÉFONO	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Figura 9.12: Opción Asignar Expresión Regular a Categoría de Entidad

Esta funcionalidad es accesible desde la página *expresiones.html* a través de las opciones en las tablas de asignación.

Esta opción permite al usuario asignar cada expresión regular a una de las categorías de entidad existentes en el proyecto.

*Nota:* debe tenerse en cuenta que una expresión regular solo podrá ser asignada a una única categoría de entidad. En caso contrario, al encontrar el patrón de coincidencia definido sobre un texto se generaría una disyuntiva acerca de qué categoría corresponde al patrón encontrado.

- **Exportar Fichero de Expresiones Regulares:**

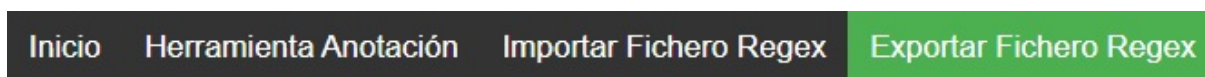


Figura 9.13: Opción Exportar Fichero de Expresiones Regulares

Esta funcionalidad es accesible desde la página *expresiones.html* a través de una opción en la barra de navegación superior.

Esta opción permite al usuario exportar un fichero con las expresiones regulares definidas en el proyecto.

*Nota:* el formato en el que se exporta el fichero de expresiones es un fichero JSON propietario de la herramienta.



Figura 9.14: Opción Importar Fichero de Expresiones Regulares

- **Importar Fichero de Expresiones Regulares:**

Esta funcionalidad es accesible desde la página *expresiones.html* a través de una opción en la barra de navegación superior.

Esta opción permite al usuario importar un fichero con expresiones regulares para ser introducidas en la interfaz de usuario de una forma directa.

*Nota:* la extensión y formato del fichero de expresiones regulares a importar debe ser un JSON propietario de la herramienta.

### Anotación Manual

Dentro de la Anotación Manual se engloban todas las funcionalidades básicas que el usuario puede realizar en la página *herramienta.html* para distintos propósitos. Este bloque incluye las capacidades fundamentales que debe tener una herramienta de anotación de textos simple, sin entrar en consideraciones de asistencia a la anotación automática.

Las funcionalidades incluidas en este bloque son las siguientes:

- **Crear Anotación:**

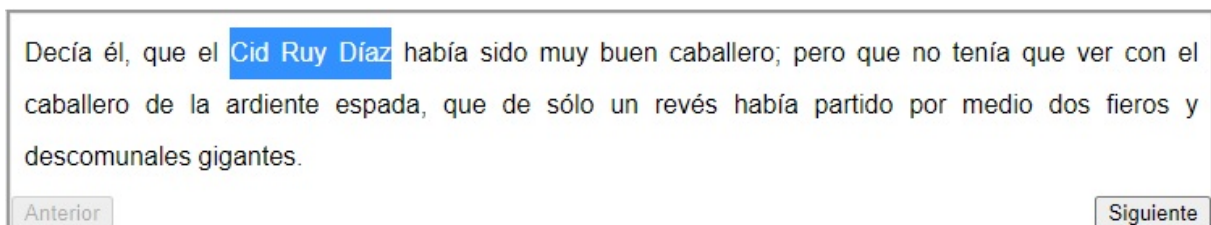


Figura 9.15: Opción Crear Anotación

Esta funcionalidad es accesible desde la página *herramienta.html* a través de una selección en el texto realizada con el ratón.

Esta opción permite al usuario crear una nueva anotación etiquetada con la categoría de entidad que esté seleccionada.

*Nota:* una anotación solo será confirmada en caso de ser válida. Selecciones en el texto que comprometan tokens (cada parte unitaria del texto) incompletas o anotaciones que se solapen con otras existentes no serán válidas.

- **Eliminar Anotación:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través del uso de ratón, el click derecho, sobre la anotación o bien a través de un icono de papelera al margen de la anotación en el panel lateral.

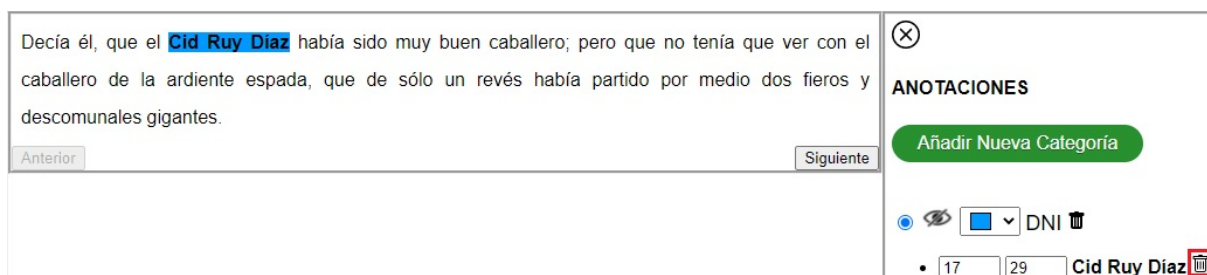


Figura 9.16: Opción Eliminar Anotación

Esta opción permite al usuario crear una nueva anotación etiquetada con la categoría de entidad que esté seleccionada.

*Nota:* una anotación solo será confirmada en caso de ser válida. Selecciones en el texto que involucren tokens (cada parte unitaria del texto) incompletos o anotaciones que se solapen con otras existentes no serán válidas.

### ■ Editar Límites de Anotación:

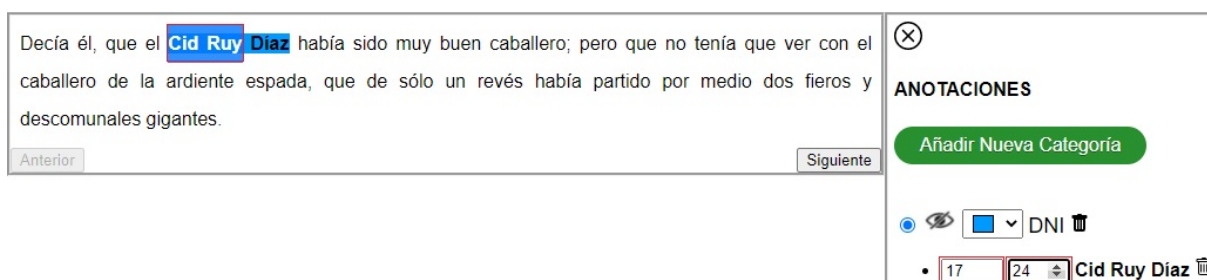


Figura 9.17: Opción Editar Límites de Anotación

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la modificación de campos numéricos asociadas a la anotación correspondiente.

Esta opción permite al usuario modificar los límites de una nueva anotación para añadir nuevos tokens o para quitarlos.

*Nota:* los límites modificados de la anotación solo serán confirmados en caso de ser válidos. Nuevos límites que involucren tokens (cada parte unitaria del texto) incompletos o que propicien el solapamiento entre anotaciones no serán válidos.

### ■ Cambiar Categoría de una Anotación:

Esta funcionalidad es accesible desde la página *herramienta.html* a través del uso avanzado de ratón, el click derecho, sobre el nombre de una anotación en la vista lateral derecha de la interfaz.

Esta opción muestra al usuario un menú emergente para cambiar seleccionar la nueva categoría de asignación a la que asociar la anotación en cuestión.





Figura 9.18: Opción Cambiar Categoría de Entidad de una Anotación

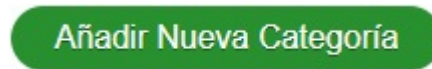


Figura 9.19: Opción Crear Categoría de Entidad

#### ■ Crear Categoría de Entidad:

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Añadir Nueva Categoría.

Esta opción permite al usuario crear una nueva categoría de entidad en el proyecto de anotación. Al seleccionar la opción se muestra un campo de texto para introducir el nombre de la categoría y un campo seleccionable de color para la nueva categoría de entidad.

*Nota:* debe tenerse en cuenta que ni el nombre ni el color de la categoría de entidad pueden coincidir con los de otra categoría de entidad existente en el proyecto. Por defecto, el color inicial establecido para la nueva categoría de entidad no estará asignado a otra categoría de entidad.

#### ■ Eliminar Categoría de Entidad:



Figura 9.20: Opción Eliminar Categoría de Entidad

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción representada por el icono de una papelera al margen de la categoría de entidad correspondiente.

Esta opción permite al usuario eliminar una categoría de entidad del proyecto de anotación actual.

*Nota:* en caso de tener asociada alguna anotación la categoría de entidad a eliminar, se pedirá primero al usuario una confirmación de la eliminación de la categoría de entidad.

#### ■ Editar Nombre de Categoría de Entidad:

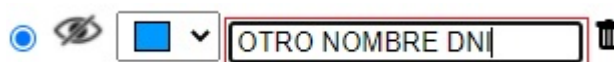


Figura 9.21: Opción Editar Nombre de Categoría de Entidad

Esta funcionalidad es accesible desde la página *herramienta.html* a través de una opción avanzada de ratón, el click derecho, sobre el nombre de una categoría de entidad.

Esta opción permite al usuario editar el nombre de la categoría de entidad en cuestión.

*Nota:* debe tenerse en cuenta que el nuevo nombre de la categoría de entidad no puede coincidir con el de otra categoría de entidad existente en el proyecto.

■ **Editar Color de Categoría de Entidad:**



Figura 9.22: Opción Editar Color de Categoría de Entidad

Esta funcionalidad es accesible desde la página *herramienta.html* a través del click izquierdo de ratón sobre el color asociado a una categoría de entidad.

Esta opción permite al usuario editar el color de la categoría de entidad en cuestión.

*Nota:* debe tenerse en cuenta que el nuevo color asociado a la categoría de entidad no puede coincidir con el asociado a otra categoría de entidad existente en el proyecto.

■ **Seleccionar Categoría de Entidad:**



Figura 9.23: Opción Seleccionar Categoría de Entidad

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción de selección al margen izquierdo de una categoría de entidad.

Esta opción permite al usuario seleccionar una categoría de entidad para el etiquetado de las siguientes anotaciones con ella.

■ **Importar Documentos de Texto:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Importar Textos dentro del desplegable de la opción contenedor Importar.

Esta opción permite al usuario importar a la herramienta ficheros de texto plano que serán cargados en la interfaz de usuario para su anotación.



Figura 9.24: Opción Importar Documentos de Texto

*Nota:* la extensión de los ficheros a importar por el usuario debe ser .txt.

■ **Importar Fichero PDF:**



Figura 9.25: Opción Importar Fichero PDF

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Importar PDF dentro del desplegable de la opción contenedor Importar.

Esta opción permite al usuario importar a la herramienta un fichero PDF para su procesamiento por parte del motor de Reconocimiento Óptico de Caracteres y posterior carga del texto resultante en la interfaz de usuario de cara a su anotación.

*Nota:* la extensión del fichero a importar por el usuario debe ser .pdf.

■ **Exportar Documentos Anotados:**

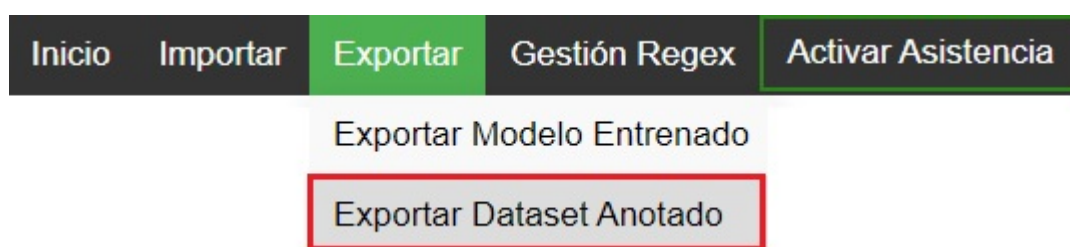


Figura 9.26: Opción Exportar Documentos Anotados

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Exportar Dataset dentro del desplegable de la opción contenedor Exportar.

Esta opción permite al usuario exportar un fichero con el dataset de documentos anotados en el proyecto. El formato de exportación puede ser decidido por el usuario entre 2 opciones posibles: formato IOB2 o formato propietario de Spacy.

■ **Importar Documentos Anotados:**



Figura 9.27: Opción Importar Documentos Anotados

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Importar Dataset dentro del desplegable de la opción contenedor Importar.

Esta opción permite al usuario importar un fichero con un dataset de documentos anotados para su carga en la interfaz de usuario de cara a su anotación.

*Nota:* la extensión del fichero a importar por el usuario debe ser `.txt` o `.spacy`.

■ **Navegar entre Documentos:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de las opciones Anterior y Siguiente bajo la vista lateral izquierda de resumen de sentencias del documento actual.

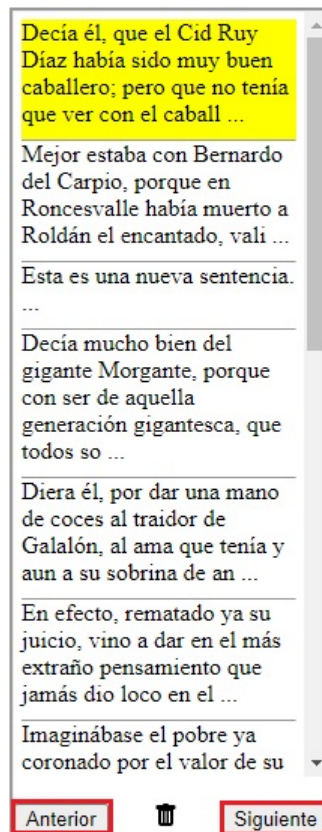


Figura 9.28: Opción Navegar entre Documentos

Esta opción permite al usuario navegar entre los documentos importados en el proyecto de anotación para cargar en la vista lateral izquierda los resúmenes de sus sentencias y cambiar el texto anotable de la vista central.

- **Navegar entre Sentencias de Documento:**

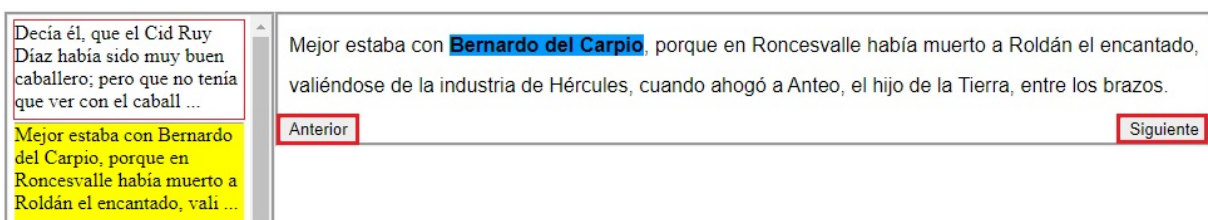


Figura 9.29: Opción Navegar entre Sentencias de Documento

Esta funcionalidad es accesible desde la página *herramienta.html* a través de las opciones Anterior y Siguiente bajo la vista central que contiene el texto actual anotable o bien a través del click izquierdo sobre alguna de las sentencias del documento presentes en la vista lateral izquierda.

Esta opción permite al usuario navegar entre las sentencias del documento actual para cargar en la vista central el texto asociado a la sentencia de cara a su anotación.

■ **Visualizar Estadísticas de Anotación:**



Figura 9.30: Opción Visualizar Estadísticas de Anotación

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Estadísticas Anotación de la parte inferior de la página.

Esta opción permite al usuario visualizar en tiempo real estadísticas de anotación del proyecto, tales como el número de documentos importados en el proyecto de anotación, de categorías de entidad, de anotaciones realizadas, de anotaciones por categoría o el ranking de categorías con más anotaciones asociadas.

■ **Establecer Opciones de Importación:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Opciones Importación dentro del desplegable de la opción contenedor Importar.

Esta opción permite al usuario establecer parámetros de importación de documentos, tales como el número de sentencias simultáneas que compondrán cada fragmento de documento o los porcentajes de recorte sobre los márgenes a aplicar en las páginas de los ficheros PDF importados a la herramienta.





Figura 9.31: Opción Establecer Opciones de Importación

### Asistencia a la Anotación

Dentro de la Asistencia a la Anotación se incluyen los procesos relacionados con los diferentes modos de asistencia a la anotación que el usuario puede ejecutar para agilizar la anotación de sus documentos.

Las funcionalidades incluidas en este bloque son las siguientes:

- **Exportar modelo de aprendizaje entrenado:**



Figura 9.32: Opción Exportar Modelo de Aprendizaje Entrenado

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Exportar Modelo Entrenado dentro del desplegable de la opción contenedor Exportar.

Esta opción permite al usuario exportar un fichero ZIP con el modelo de aprendizaje automático entrenado durante la etapa de asistencia a la anotación del proyecto de anotación.

- **Importar modelo de aprendizaje preentrenado:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Importar Modelo dentro del desplegable de la opción contenedor Importar.

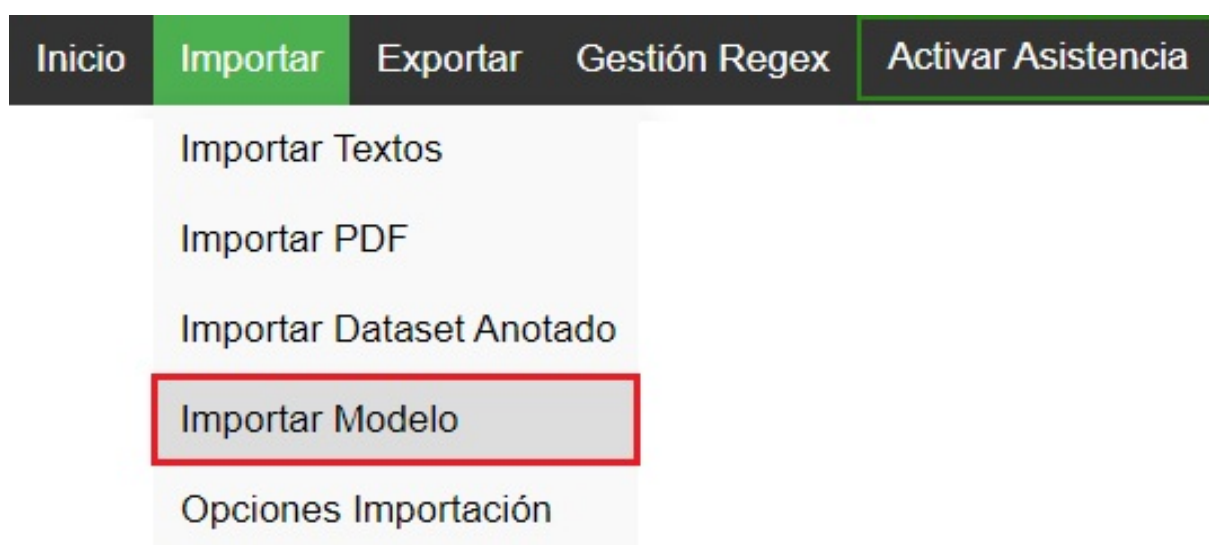


Figura 9.33: Opción Importar Modelo de Aprendizaje Preentrenado

Esta opción permite al usuario importar un fichero ZIP con un modelo de aprendizaje automático preentrenado de cara a aplicar sus predicciones en la etapa de Asistencia a la Anotación con Predicciones de Modelo Importado.

*Nota:* la extensión del fichero a importar por el usuario debe ser .zip.

■ **Activar asistencia con patrones de coincidencia:**



Figura 9.34: Opción Activar Asistencia con Patrones de Coincidencia

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Asistencia con Patrones dentro de la opción Activar Asistencia.

Esta opción permite al usuario activar la asistencia a la anotación mediante el uso de patrones de coincidencia. Los resultados obtenidos de este proceso serán incluidos en la interfaz de usuario de la herramienta.

*Nota:* este modo de asistencia a la anotación solo puede activarse en caso de tener alguna expresión regular asignada a una categoría de entidad en el proyecto.



- **Activar asistencia con entrenamiento de modelo de aprendizaje y patrones de coincidencia:**

Figura 9.35: Opción Activar Asistencia con Entrenamiento de Modelo de Aprendizaje y Patrones de Coincidencia

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Asistencia con Entrenamiento de Modelo + Patrones dentro de la opción Activar Asistencia.

Esta opción permite al usuario activar la asistencia a la anotación mediante el entrenamiento de un modelo de aprendizaje con las anotaciones realizadas en los documentos del proyecto junto con el uso de patrones de coincidencia. Los resultados obtenidos de este proceso serán incluidos en la interfaz de usuario de la herramienta.

*Nota:* este modo de asistencia a la anotación solo puede activarse en caso de tener anotaciones realizadas en los documentos del proyecto con las que poder entrenar el modelo de aprendizaje.

- **Activar asistencia con predicciones de modelo de aprendizaje importado y patrones de coincidencia:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Asistencia con Modelo Importado dentro de la opción Activar Asistencia.

Esta opción permite al usuario activar la asistencia a la anotación mediante las predicciones del modelo de aprendizaje importado en el proyecto junto con el uso de patrones de coincidencia. Los resultados obtenidos de este proceso serán incluidos en la interfaz de usuario de la herramienta.

*Nota:* este modo de asistencia a la anotación solo puede activarse en caso de tener un modelo de aprendizaje importado en el proyecto.

- **Visualizar resultados del entrenamiento de un modelo de aprendizaje:**

Esta funcionalidad es accesible desde la página *herramienta.html* a través de la opción Resultados Entrenamiento de la parte inferior de la página. Esta opción permite al usuario

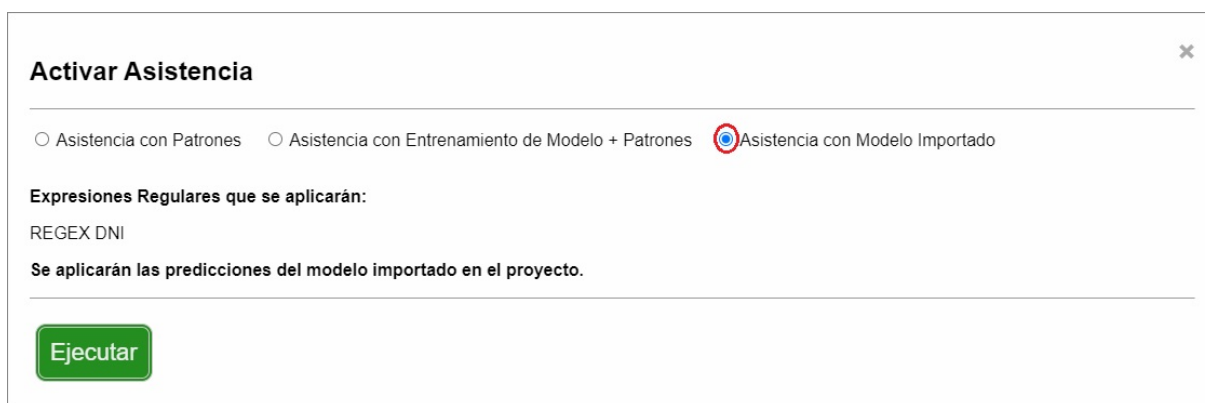


Figura 9.36: Opción Activar Asistencia con Predicciones de Modelo de Aprendizaje Importado

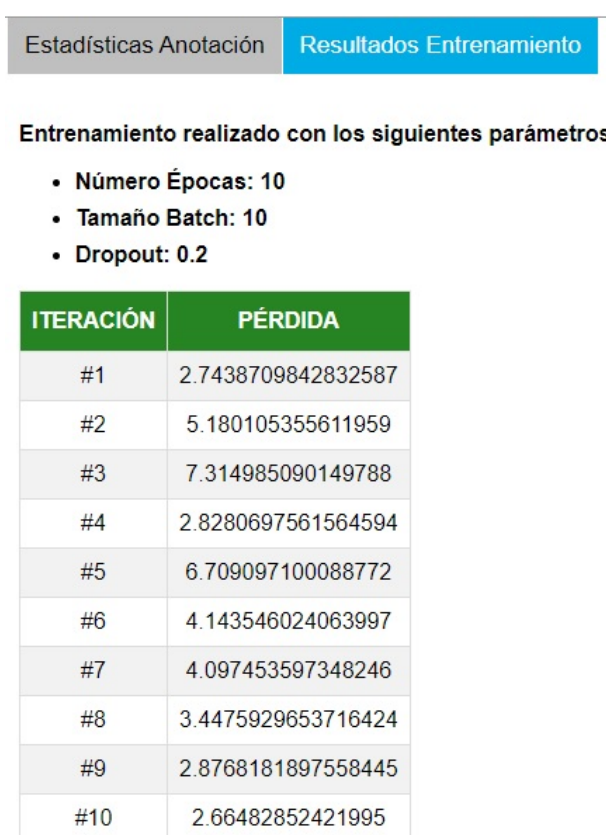


Figura 9.37: Opción Visualizar Resultados de Entrenamiento

visualizar resultados del entrenamiento derivados de la etapa de asistencia a la anotación mediante el entrenamiento de un modelo de aprendizaje a partir de las anotaciones realizadas en los documentos del proyecto junto con el uso de patrones de coincidencia. Entre la información mostrada se encuentran los parámetros con los que se entrenó el modelo y los valores de pérdida asociados a cada iteración del entrenamiento realizado.

# Capítulo 10

## Conclusiones y Trabajo Futuro

En este capítulo del proyecto se exponen unas conclusiones generales derivadas del trabajo realizado durante estos meses de proyecto. Además, se presentan varias líneas de trabajo futuro por las que continuar con la implementación del producto.

### 10.1. Conclusiones

A nivel de proyecto, podemos concluir que se ha trabajado de forma exitosa con la metodología ágil UVagile y se han conseguido los objetivos marcados dentro del alcance inicial que planteamos para este proyecto. El desarrollo con una metodología ágil ha permitido ir acometiendo el proyecto de manera incremental por Sprints, una tendencia más actual en el desarrollo de software en la industria.

A nivel de producto, se ha logrado implementar una herramienta útil para la anotación de entidades nombradas en textos de forma manual, que además, contara con un mecanismo de asistencia a la anotación que agilizará el proceso de anotación a realizar por el usuario. Adicionalmente, la herramienta ha podido ser aplicada al contexto de un caso de uso real con datos aportados por el Ayuntamiento de Valladolid.

A nivel personal, la realización de este proyecto me ha permitido tener una mayor perspectiva de lo que supone la realización de un proyecto desde cero, abarcando todas las fases del mismo, algo muy distinto a la realización de una práctica guionizada. He asumido una mayor autonomía a la hora de realizar el trabajo, siendo también asesorado por mis tutores en lo que necesitara, y he explorado la investigación de nuevas tecnologías que no había tratado durante el grado. Ello ha supuesto un esfuerzo adicional por mi parte en cambiar la manera de pensar y de afrontar el proyecto en su conjunto.

Dentro de las competencias personales he potenciado algunas como la capacidad de análisis, la resolución de problemas, la toma de decisiones, la capacidad de aplicar los conocimientos aprendidos en la práctica, las habilidades de investigación o la habilidad para trabajar de forma autónoma, entre otras.

Por otra parte, desde el punto de vista técnico, he aprendido nuevas tecnologías y mejorado en el conocimiento que tenía de otras. Al ser este un proyecto con desarrollo web, he podido

mejorar mis conocimientos y soltura en tecnologías web básicas como HTML, CSS o Javascript. También, he aprendido a trabajar con un framework web de actualidad, como es Django, para la creación de sitios web dinámicos desarrollando el backend de la aplicación en lenguaje Python. De hecho, en Python se han implementado las soluciones relativas al Reconocimiento de Entidad Nombradas, una tarea dentro del Procesamiento del Lenguaje Natural, un campo en auge dentro de la Inteligencia Artificial. A su vez, se ha trabajado con una propuesta distinta para la creación de documentación técnica de alta calidad, como es LaTeX. Ello también requerido de un aprendizaje previo para su uso.

En definitiva, creo que la realización de este proyecto de Trabajo de Fin de Grado ha supuesto para mí una oportunidad para aprender nuevas tecnologías que desconocía y poder aplicar los conocimientos adquiridos a lo largo del grado en la realización de un proyecto autónomo desde cero.

## 10.2. Trabajo Futuro

Las líneas de trabajo futuro representan las oportunidades para continuar la implementación del producto software construido, tanto a nivel de nueva funcionalidad como de mejoramiento o extensión de las existentes.

Este proyecto presenta varias líneas de trabajo que motivan su futura continuidad. El alcance de nuestro producto *Annotator* se puede ver ampliado con algunas líneas de trabajo futuro que podrían considerarse para una próxima versión del mismo.

Algunas de estas líneas trabajo futuro que se proponen son las siguientes:

1. **Despliegue web de la herramienta:** la primera posible línea de trabajo futuro sería desplegar la herramienta en servidores de producción para su accesibilidad web por parte de diferentes usuarios concurrentes. Al haber desarrollado la herramienta como una aplicación web, este proceso se vería facilitado.

Para llevar a cabo el despliegue web de la herramienta a un entorno de producción, se deberían considerar nuevos requisitos no funcionales relativos a la disponibilidad, la seguridad, la fiabilidad o la escalabilidad, entre otros. Ante la inclusión de estos nuevos requisitos no funcionales, la arquitectura física general del sistema se vería modificada.

Igualmente, el servidor web en que se desplegara la aplicación sería distinto del ofrecido por Django para el propio desarrollo de la aplicación. Por ejemplo, podría adoptarse la medida de desplegar la aplicación en un servidor de aplicaciones como Gunicorn.

La siguiente propuesta es una aproximación que se podría contemplar para cumplir con estas nuevas consideraciones. En ella se utilizan servidores de aplicación Gunicorn para servir la aplicación web y se establecen 2 proxy inversos Nginx para el balanceo de carga entre los servidores Gunicorn. Esta arquitectura soporta el cumplimiento de los requisitos no funcionales mencionados anteriormente. A su vez, se podría utilizar una base de datos Oracle para la parte de almacenamiento de los proyectos de anotación.

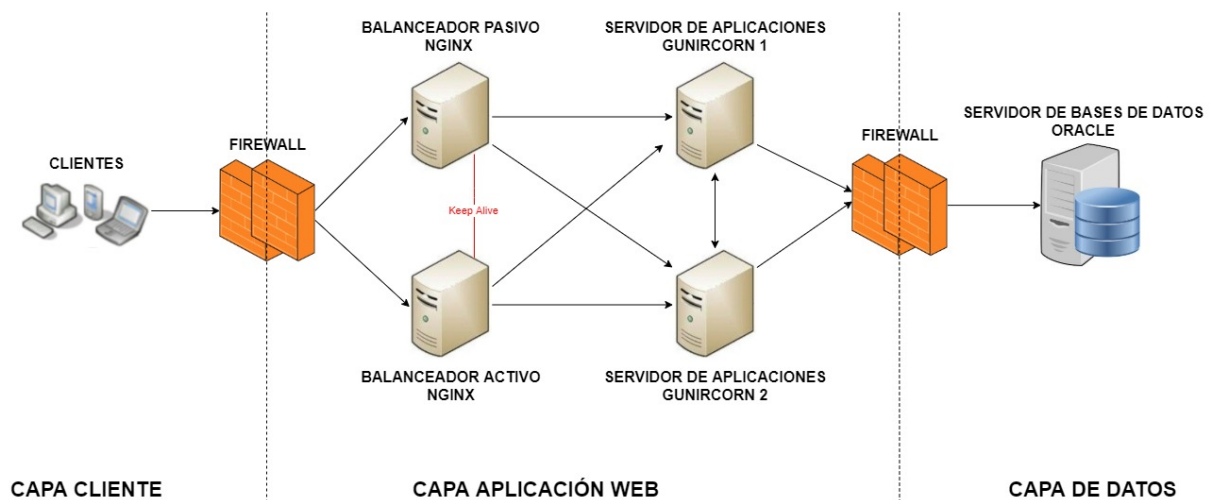


Figura 10.1: Arquitectura Física propuesta para el despliegue de la herramienta en producción

2. **Incluir la evaluación de modelos de aprendizaje automático:** esta funcionalidad consistirá en la posibilidad por parte del usuario de subir a la herramienta varios modelos de aprendizaje automático preentrenados y evaluar su desempeño en la extracción de las entidades nombradas en documentos de prueba preanotados. Una vez realizada la evaluación de los modelos, se podría exportar al cliente un fichero con los resultados de la evaluación para una posterior interpretación.
3. **Soporte para la exposición de las funcionalidades de la herramienta como servicios web:** consistiría en ofrecer ciertas funcionalidades de la herramienta como servicios web para su uso desde otras aplicaciones. Para ello, se establecerían vistas en Django mapeadas a URL para ejecutar la funcionalidad requerida y devolver una respuesta en un formato como podría ser JSON.

Una utilidad interesante para ser expuesta sería un servicio web REST para derivar las entidades nombradas contenidas en un texto de entrada, abstrayendo a la aplicación que realiza la petición al servicio web de toda la lógica de negocio subyacente.

4. **Añadir la gestión de cuentas de usuario:** podría ser interesante valorar la opción de gestionar cuentas de usuario dentro de la herramienta y asociarlas a proyectos de anotación privados a dichas cuentas. Esta característica sería especialmente interesante ante el despliegue web de la herramienta.
5. **Internacionalización de la herramienta:** esta característica permitiría cambiar el idioma en el que están descritos los componentes de la interfaz de usuario, así como los mensajes de advertencia y error que se muestran como guía al usuario de la herramienta.
6. **Ampliar la visión de producto para contemplar la extracción de relaciones entre entidades:** esta es otra tarea dentro del Procesamiento del Lenguaje Natural consistente en hallar las relaciones existentes entre las entidades nombradas de un texto. Esta tarea aporta

mayor información que el simple Reconocimiento de Entidades Nombradas, pues se basa en la creación de etiquetas entre entidades nombradas que estén íntimamente relacionadas en torno a algún hecho o condición determinada. A su vez, el conocimiento de estas relaciones podría hacer mejorar el desempeño del propio Reconocimiento de Entidades Nombradas.

7. **Añadir una nueva vista para visualizar todas las anotaciones de entidades derivadas por un modelo de aprendizaje:** esta nueva vista listaría por categorías de entidad todas las entidades nombradas extraídas por el modelo y podría permitir al usuario visualizar de un solo vistazo las entidades de una misma categoría que ha obtenido el modelo de aprendizaje.

Una de sus utilidades de interés sería la evaluación del motor OCR utilizado para digitalizar textos de entrada. Consistiría en detectar palabras con erratas que el modelo de aprendizaje ha podido clasificar bien, como si fueran la misma entidad que su forma correcta, pero que el motor OCR ha transcrito de forma errónea en el proceso de Reconocimiento Óptico de Caracteres. Este estudio podría desvelar fallos en el proceso general de digitalización de documentos que podrían ser tratados para lograr subsanarlos.

8. **Añadir la capacidad de categorización de textos:** esta característica consistiría en la predicción de una categoría global con la que clasificar cada texto importado por el usuario a la aplicación. También, podría sugerirse una categoría global para todos los documentos del proyecto. Esta funcionalidad podría resolverse añadiendo un componente de Spacy al pipeline actual. En concreto, este componente recibe el nombre de TextCategorizer.

La utilidad de esta funcionalidad radica en la posibilidad de poder categorizar un dataset completo de documentos y, en consecuencia, poder determinar un modelo de NER adecuado que pudiera funcionar de forma eficiente en la extracción de las entidades nombradas del tipo de documentos en cuestión.

# Bibliografía

- [1] Miguel A. Martínez-Prieto, Jorge Silvestre, Anibal Bregon y José Ignacio Farrán. *Hacia la consolidación de las aulas ágiles*. Actas de las XXVI Jornadas sobre Enseñanza Universitaria de la Informática Volumen 5, págs. 29-36, 2020.
- [2] Ken Schwaber y Jeff Sutherland. *La Guía Scrum*. Scrum.org, 2020.
- [3] Graham Wilcock. *Introduction to linguistic annotation and text analytics*. Synthesis Lectures on Human Language Technologies 2(1), págs. 1-159, 2009.
- [4] Gobinda G. Chowdhury. *Natural Language Processing*. Annual Review of Information Science and Technology 37(1), págs. 51-89, 2003.
- [5] Behrang Mohit. *Named Entity Recognition*. Natural Language Processing of Semitic Languages, 2014.
- [6] Arindam Chaudhuri, Krupa Mandaviya, Pratixa Badelia y Soumya K. Ghosh. *Optical Character Recognition Systems*. Optical Character Recognition Systems for Different Languages with Soft Computing Volumen 352, 2017.
- [7] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. *Deep learning*. Nature Volumen 352, págs. 436–444, 2015.
- [8] Nasser Alshammari y Saad Alanazi. *The impact of using different annotation schemes on named entity recognition*. Egyptian Informatics Journal, 2020.
- [9] Wei-Te Chen y Will Styler. *Anafora: A Web-based General Purpose Annotation Tool*. Proceedings of the 2013 NAACL HLT Demonstration Session, págs. 14-19, 2013.
- [10] *Text Annotation in Machine Learning*. Kili Technology. Abril 2021. URL: <https://kili-technology.com/blog/text-annotation-in-machine-learning-an-overview/>
- [11] *Text Annotation Tools in 2020*. Bohemian AI. Abril 2021. URL: <https://bohemian.ai/blog/text-annotation-tools-which-one-pick-2020/>
- [12] *Guide to Natural Language Processing (NLP)*. Towards Data Science. Abril 2021. URL: <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>

- [13] *What is named entity recognition (NER)?*. Medium. Abril 2021. URL: <https://medium.com/mysuperaai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d>
- [14] *Prodigy 101 – everything you need to know*. Prodigy. Abril 2021. URL: <https://prodigy/docs>
- [15] *Welcome to doccano*. Doccano. Abril 2021. URL: <https://doccano.github.io/doccano/>
- [16] *brat rapid annotation tool*. Brat. Abril 2021. URL: <https://brat.nlplab.org/>
- [17] *Welcome to tagtog*. Tagtog. Abril 2021. URL: <https://docs.tagtog.net/>
- [18] *Planning Poker*. Mountain Goat Software. Abril 2021. URL: <https://www.mountaingoatsoftware.com/agile/planning-poker>
- [19] *Clase Selection Javascript*. MDN Web Docs. Mayo 2021. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Selection>
- [20] *Clase Range Javascript*. MDN Web Docs. Mayo 2021. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Range>
- [21] *Tutorial Uso Selection y Range*. Javascript.info. Mayo 2021. URL: <https://es.javascript.info/selection-range>
- [22] *Tutoriales Framework Web Django (Python)*. MDN Web Docs. Mayo 2021. URL: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/>
- [23] *Django Documentation*. Django. Junio 2021. URL: <https://docs.djangoproject.com/en/3.2/>
- [24] *Handling Ajax request in Django*. Geeks for Geeks. Mayo 2021. URL: <https://www.geeksforgeeks.org/handling-ajax-request-in-django/>
- [25] *Tutoriales HTML, CSS y Javascript*. W3Schools. Junio 2021. URL: <https://www.w3schools.com/>
- [26] *jQuery Ajax*. jQuery. Mayo 2021. URL: <https://api.jquery.com/jquery.ajax/>
- [27] *Real Python Tutorials*. Real Python. Junio 2021. URL: <https://realpython.com/>
- [28] *Regular Expression Operations*. Python. Mayo 2021. URL: <https://docs.python.org/3/library/re.html>
- [29] *Spacy docs*. Spacy. Mayo 2021. URL: <https://spacy.io/>
- [30] *Kaggle Datasets and Code*. Kaggle. Junio 2021. URL: <https://www.kaggle.com/>



- 
- [31] *A Quick Guide to Tokenization, Lemmatization, Stop Words and Phrase Matching using Spacy*. Data Science Duniya. Junio 2021. URL: <https://ashutoshtriplathi.com/2020/04/06/guide-to-tokenization-lemmatization-stop-words-and-phrase-matching-using-spacy/>
- [32] *How to Train NER with Custom training data using spaCy*. Manivannan Murugavel - Medium. Junio 2021. URL: <https://manivannan-ai.medium.com/how-to-train-ner-with-custom-training-data-using-spacy-188e0e508c6>
- [33] *Named Entity Recognition (NER) using spaCy*. Towards Data Science. Junio 2021. URL: <https://towardsdatascience.com/named-entity-recognition-ner-using-spacy-nlp-part-4-28da2ece57c6>
- [34] *Named Entity Recognition (NER) with Spacy and Python*. Manivannan Murugavel - IT-NEXT. Junio 2021. URL: <https://itnext.io/nlp-named-entity-recognition-ner-with-spacy-and-python-dabaf843cab2>
- [35] *Building a Flask API to Automatically Extract Named Entities Using SpaCy*. Towards Data Science. Junio 2021. URL: <https://towardsdatascience.com/building-a-flask-api-to-automatically-extract-named-entities-using-spacy-2fd3f54ebbc6>
- [36] *How to Run Linux Commands with Python*. Circuit Basics. Junio 2021. URL: <https://www.circuitbasics.com/run-linux-commands-with-python/>
- [37] *A Beginner's Guide to Tesseract OCR*. Better Programming. Junio 2021. URL: <https://betterprogramming.pub/beginners-guide-to-tesseract-ocr-using-python-10ecbb426c3d>
- [38] *Librería de Python Pytesseract*. PyPI. Junio 2021. URL: <https://pypi.org/project/pytesseract/>
- [39] *PyTesseract: Simple Python Optical Character Recognition*. Stack Abuse. Junio 2021. URL: <https://stackabuse.com/pytesseract-simple-python-optical-character-recognition>
- [40] *Reading contents of PDF using OCR (Optical Character Recognition)*. Geeks for Geeks. Junio 2021. URL: <https://www.geeksforgeeks.org/python-reading-contents-of-pdf-using-ocr-optical-character-recognition/>