



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**IAGScore
Documentación Técnica**



Presentado por Pedro Antonio Abellaneda
Canales
en Universidad de Burgos — 10 de mayo
de 2025

Tutor: Raúl Marticorena Sánchez

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	3
Apéndice B Especificación de Requisitos	5
B.1. Introducción	5
B.2. Objetivos generales	5
B.3. Catálogo de requisitos	6
B.4. Especificación de requisitos	7
Apéndice C Especificación de diseño	15
C.1. Introducción	15
C.2. Diseño de datos	15
C.3. Diseño arquitectónico	16
C.4. Diseño procedimental	16
Apéndice D Documentación técnica de programación	17
D.1. Introducción	17
D.2. Estructura de directorios	17
D.3. Manual del programador	22

D.4. Compilación, instalación y ejecución del proyecto	22
D.5. Pruebas del sistema	22
Apéndice E Documentación de usuario	23
E.1. Introducción	23
E.2. Requisitos de usuarios	23
E.3. Instalación	23
E.4. Manual del usuario	23
Apéndice F Anexo de sostenibilización curricular	25
F.1. Introducción	25
Bibliografía	27

Índice de figuras

B.1. Diagrama de caso de uso general	7
C.1. Diagrama ER	15
C.2. Diagrama de relación	16

Índice de tablas

B.1. CU-1 Registrar usuario.	8
B.2. CU-2 Login de usuario.	9
B.3. CU-3 Importar rúbrica.	10
B.4. CU-4 Listar rúbricas.	11
B.5. CU-5 Mostrar rúbrica.	11
B.6. CU-6 Eliminar rúbrica.	12
B.7. CU-7 Crear Prompts.	13
B.8. CU-8 Listar Prompts.	14

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

En la gestión del proyecto se ha optado por seguir una metodología de desarrollo ágil *Scrum* con algunas salvedades, al participar una persona con la supervisión del tutor y según la guía Scrum: *"Scrum emplea un enfoque iterativo e incremental para optimizar la previsibilidad y controlar el riesgo. Scrum involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir tales habilidades según sea necesario."* [1]

Para el desarrollo incremental del proyecto se han planificado una serie de sprints de dos semanas de duración, planificando al inicio de cada uno de ellos las tareas a desarrollar y revisando estas tareas a la finalización del mismo.

Para la gestión de los sprints, planificación y asignación de tareas se usó la herramienta *Zube*

A continuación se muestran los *sprints* llevados a cabo durante las distintas etapas de este proyecto.

Sprint 0 (27/02/2025 hasta 6/03/2025)

Se define un sprint inicial de una semana de duración en el que se realiza una primera aproximación a la propuesta de proyecto. En este sprint se realizan las siguientes tareas:

- Entrevista con tutor.
- Preparación de los entornos Github, Zube.
- Investigación sobre herramientas.
- Investigación sobre LLMs.
- Inicio de la documentación.
- Configuración del entorno de desarrollo.
- Definición de objetivos generales.
- Diseño de caso de uso genérico.

Sprint 1 (7/03/2025 hasta 20/03/2025)

Segundo sprint en el que se realiza una planificación más específica de las tareas:

- Implementación de Login.
- Creación de un mockup del esqueleto de la página Home.
- Realización de pruebas con distintos LLMs.
- Revisión y actualización del caso de uso general.
- Diseño de diagramas ER.
- Personalización del modelo de usuario.
- Implementación del registro de usuarios.

Sprint 2 (20/03/2025 hasta 3/04/2025)

A continuación se muestran las tareas planificadas:

- Implementación de Home page.
- Implementación inicial de la sección rúbricas.
- Implementación inicial de la sección prompts.
- Integración de SonarQube cloud.
- Revisión de Login y Registro de usuarios.

Sprint 3 (03/04/2025 hasta 16/04/2025)

A continuación se muestran las tareas planificadas:

- Implementación de modelo Correcciones.
- Implementación de formulario para el modelo.
- Implementación inicial de la carga de ficheros.
- Implementación de test.
- Integración de las vistas.
- Revisión de Rúbricas y Prompts.

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice *B*

Especificación de Requisitos

B.1. Introducción

Una muestra de cómo podría ser una tabla de casos de uso:

B.2. Objetivos generales

Este trabajo se ha realizado persiguiendo el cumplimiento de los siguientes objetivos:

- Desarrollar un *sistema software* que permita a los usuarios la corrección de ejercicios de programación.
- El usuario debe estar *registrado* para acceder al sistema iniciando sesión.
- Los usuarios *importarán* sus propias rúbricas en formato Markdown
- Los usuarios crearán sus propios prompts.
- La evaluación de los ejercicios se realizará mediante *modelos de lenguaje LLM*.
- Los usuarios podrán configurar aspectos determinados del LLM usado.

B.3. Catálogo de requisitos

Requisitos funcionales y no funcionales para cumplir con los objetivos generales

Requisitos funcionales

- **RF-1 - Registro de usuarios.** Los usuarios podrán registrarse mediante nombre, correo electrónico y contraseña.
- **RF-2 - Login de usuarios.** Los usuarios deberán iniciar sesión para usar la aplicación.
- **RF-3 - Importar rúbricas.** Los usuarios podrán importar sus propias rúbricas en formato Markdown.
 - **RF-3.1 - Listar rúbricas.** Los usuarios podrán consultar sus rúbricas importadas.
 - **RF-3.2 - Mostrar rubrica.** Los usuarios podrán visualizar los detalles de una rúbrica concreta.
 - **RF-3.3 - Eliminar rúbrica.** Los usuarios podrán eliminar una rúbrica importada.
- **RF-4 Crear prompts.** Los usuarios crearán sus propios prompts.
 - **RF-4.1 - Listar prompts.** Los usuarios podrán consultar sus prompts creados.
 - **RF-4.2 - Mostrar prompt.** Los usuarios podrán visualizar un prompt concreto.
 - **RF-4.3 - Eliminar prompt.** Los usuarios podrán eliminar un prompt.
- **RF-5 Configurar correcciones.** Los usuarios configurarán las correcciones.
 - **RF-5.1 - Listar correcciones.** Los usuarios podrán consultar sus correcciones configuradas.
 - **RF-5.2 - Mostrar corrección.** Los usuarios podrán visualizar un los detalles de configuración de una corrección.
 - **RF-4.3 - Eliminar corrección.** Los usuarios podrán eliminar una corrección.

- **RF-5.4 - Ejecutar corrección.** Los usuarios podrán ejecutar una corrección.
- **RF-6 - Configurar LLM.** Los usuarios podrán configurar aspectos determinados del LLM seleccionado.
 - **RF-6.1 - Seleccionar LLM.** Los usuarios seleccionarán el LLM a usar.
 - **RF-6.2 - Parametrizar LLM.** Los usuarios podrán modificar determinados parámetros del LLM seleccionado.

B.4. Especificación de requisitos

Diagramas de casos de uso y detalles sobre los mismos.

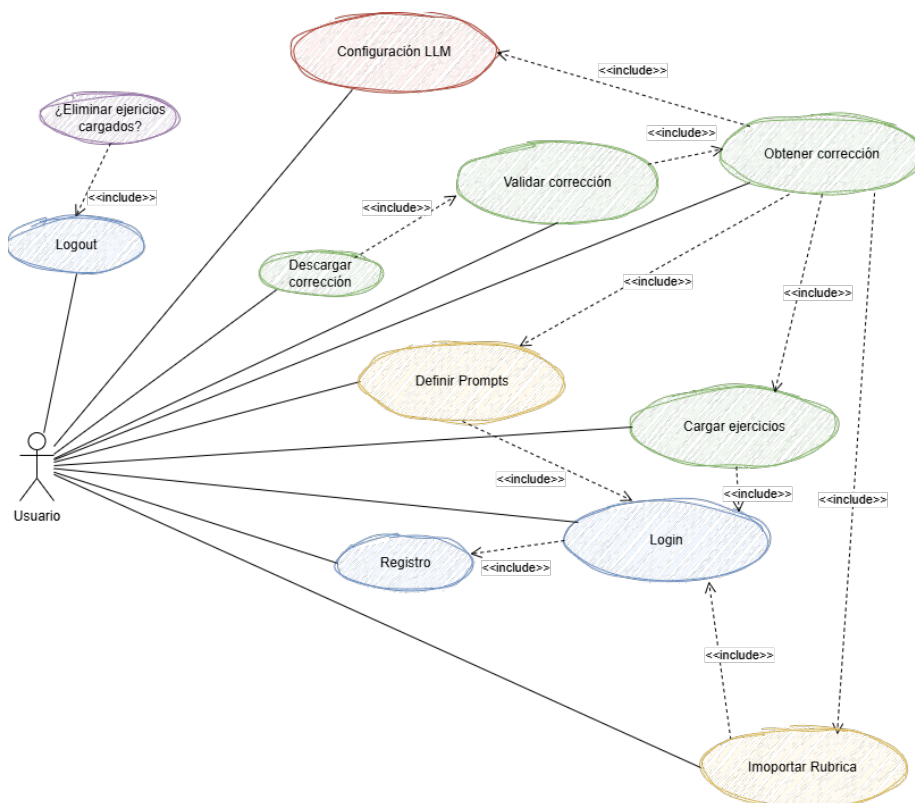
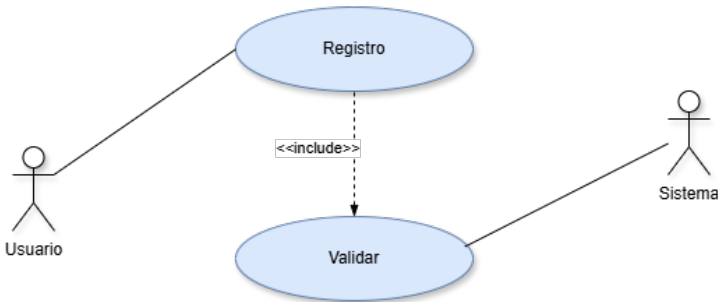
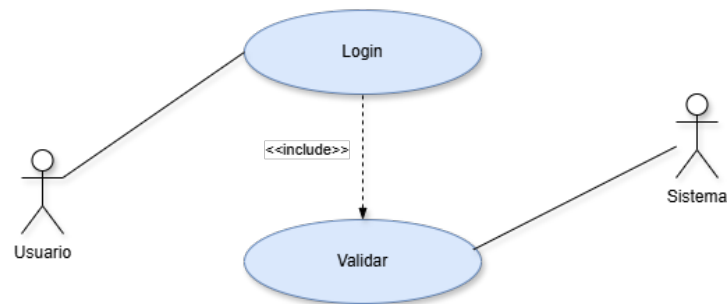


Figura B.1: Diagrama de caso de uso general



CU-1	Registro de usuario
Versión	1.0
Autor	Pedro Antonio Abellaneda Canales
Requisitos asociados	RF-1
Descripción	Permite registrarse a un usuario
Precondición	El usuario no puede estar logeado
Acciones	<div>1. El usuario entra en la aplicación y accede a la página de registro</div> <div>2. El usuario introduce su email y una contraseña</div> <div>3. El usuario confirma el registro con el botón registrar</div>
Postcondición	La aplicación redirecciona a la página de login. Mensaje: Usuario registrado correctamente.
Excepciones	Mensaje Mensaje Mensaje
Importancia	Alta

Tabla B.1: CU-1 Registrar usuario.



CU-2	Login de usuario
Versión	1.0
Autor	Pedro Antonio Abellaneda Canales
Requisitos asociados	RF-2
Descripción	Permite logearse a un usuario
Precondición	El usuario debe estar registrado
Acciones	<div>1. El usuario entra en la aplicación y accede a la página de login. 2. El usuario introduce su email y una contraseña. 3. El usuario confirma pulsando el botón entrar.</div>
Postcondición	La aplicación redirecciona a la página home.
Excepciones	Mensaje Mensaje Mensaje
Importancia	Alta

Tabla B.2: CU-2 Login de usuario.

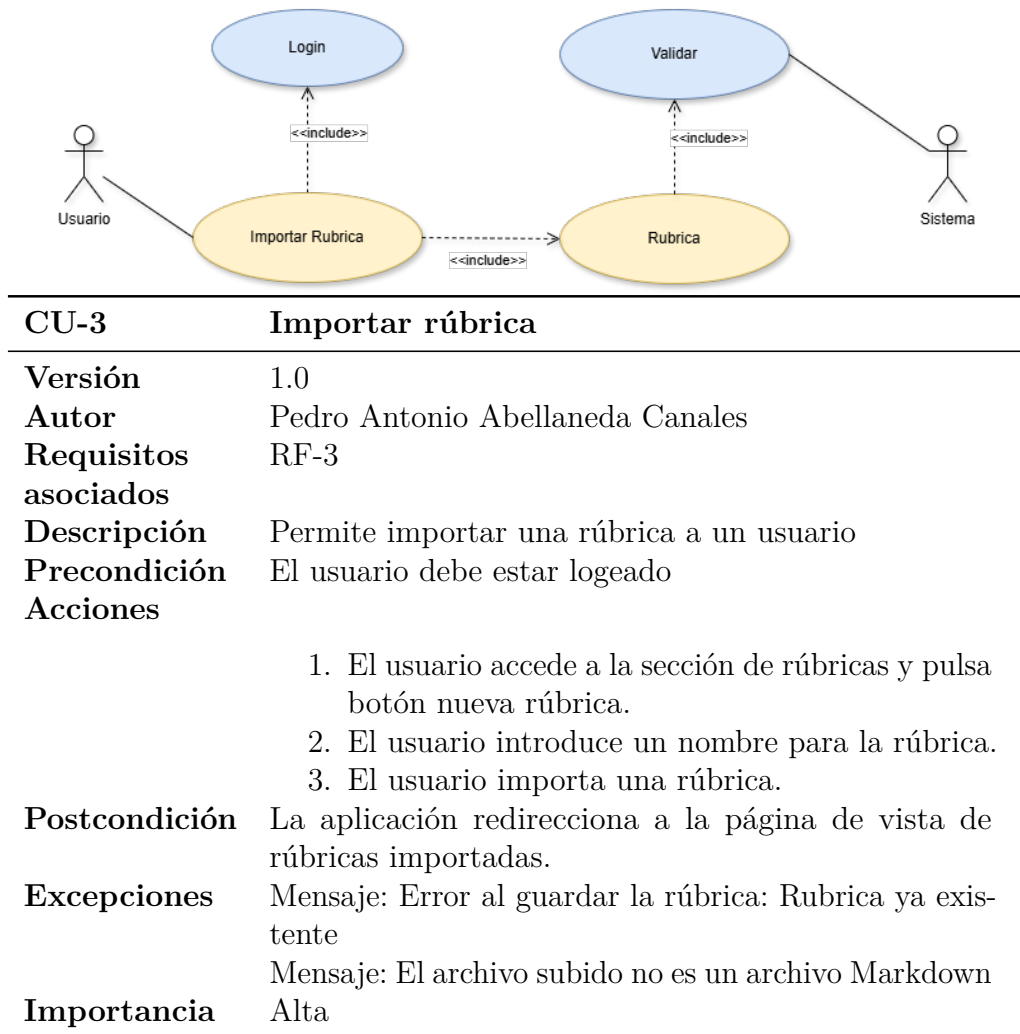


Tabla B.3: CU-3 Importar rúbrica.

CU-4	Listar rúbricas
Versión	1.0
Autor	Pedro Antonio Abellaneda Canales
Requisitos asociados	RF-3.1
Descripción	Permite ver las rúbricas del usuario
Precondición	El usuario debe estar logeado El usuario debe haber importado y creado alguna rúbrica
Acciones	
Postcondición	1. El usuario accede a la sección de rúbricas Se muestra la tabla con nombre y fecha de creación de cada una de las rúbricas creadas
Excepciones	
Importancia	Alta

Tabla B.4: CU-4 Listar rúbricas.

CU-5	Mostrar rúbrica
Versión	1.0
Autor	Pedro Antonio Abellaneda Canales
Requisitos asociados	RF-3.2
Descripción	Permite mostrar una rúbrica a un usuario
Precondición	El usuario debe estar logeado La rúbrica debe estar creada por el usuario
Acciones	
Postcondición	1. El usuario accede a la sección de rúbricas. 2. El usuario busca en la tabla la rúbrica que desee. 3. El usuario pulsa el botón mostrar. La aplicación redirecciona a la página de vista de una rúbrica concreta.
Excepciones	Mensaje Mensaje Mensaje
Importancia	Alta

Tabla B.5: CU-5 Mostrar rúbrica.

CU-6	Eliminar rúbrica
Versión	1.0
Autor	Pedro Antonio Abellameda Canales
Requisitos asociados	RF-3.3
Descripción	Permite eliminar una rúbrica a un usuario
Precondición	El usuario debe estar logeado La rúbrica debe estar creada por el usuario
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la sección de rúbricas. 2. El usuario busca en la tabla la rúbrica que desee. 3. El usuario pulsa el botón eliminar. 4. El usuario confirma la opción eliminar.
Postcondición	La aplicación redirecciona a la página rúbricas
Excepciones	Mensaje Mensaje Mensaje
Importancia	Alta

Tabla B.6: CU-6 Eliminar rúbrica.

CU-7	Crear Prompts
Versión	1.0
Autor	Pedro Antonio Abellaneda Canales
Requisitos asociados	RF-4
Descripción	Permite crear un prompt a un usuario
Precondición	El usuario debe estar logeado
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la sección de prompts y pulsa botón nuevo prompt. 2. El usuario introduce un nombre. 3. El usuario introduce el texto del prompt. 4. El usuario pulsa botón importar.
Postcondición	<ul style="list-style-type: none"> ■ La aplicación redirecciona a la página de prompts ■ Mensaje: Prompt creado correctamente
Excepciones	<ul style="list-style-type: none"> ■ Mensaje: ■ Mensaje:
Importancia	Alta

Tabla B.7: CU-7 Crear Prompts.

CU-8	Listar Prompts
Versión	1.0
Autor	Pedro Antonio Abellaneda Canales
Requisitos asociados	RF-4.1
Descripción	Permite listar los prompts de un usuario
Precondición	<ul style="list-style-type: none"> ■ El usuario debe estar logeado. ■ El usuario debe haber creado algún prompt.
Acciones	1. El usuario accede a la sección de prompts.
Postcondición	<ul style="list-style-type: none"> ■ Se muestra la tabla con nombre y fecha de creación de cada una de los prompts creados.
Excepciones	
Importancia	Alta

Tabla B.8: CU-8 Listar Prompts.

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

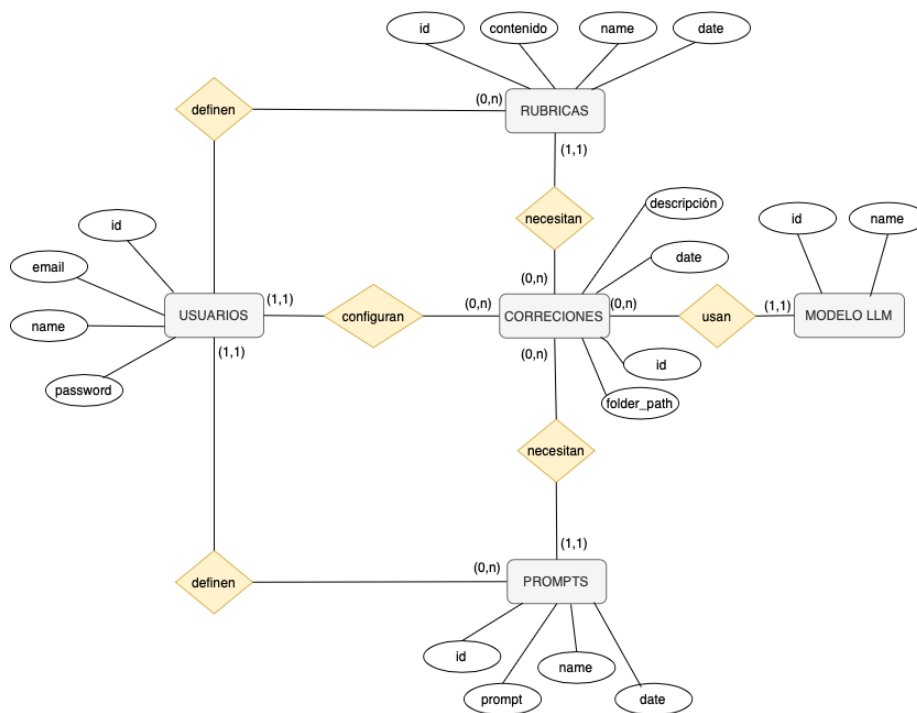


Figura C.1: Diagrama ER

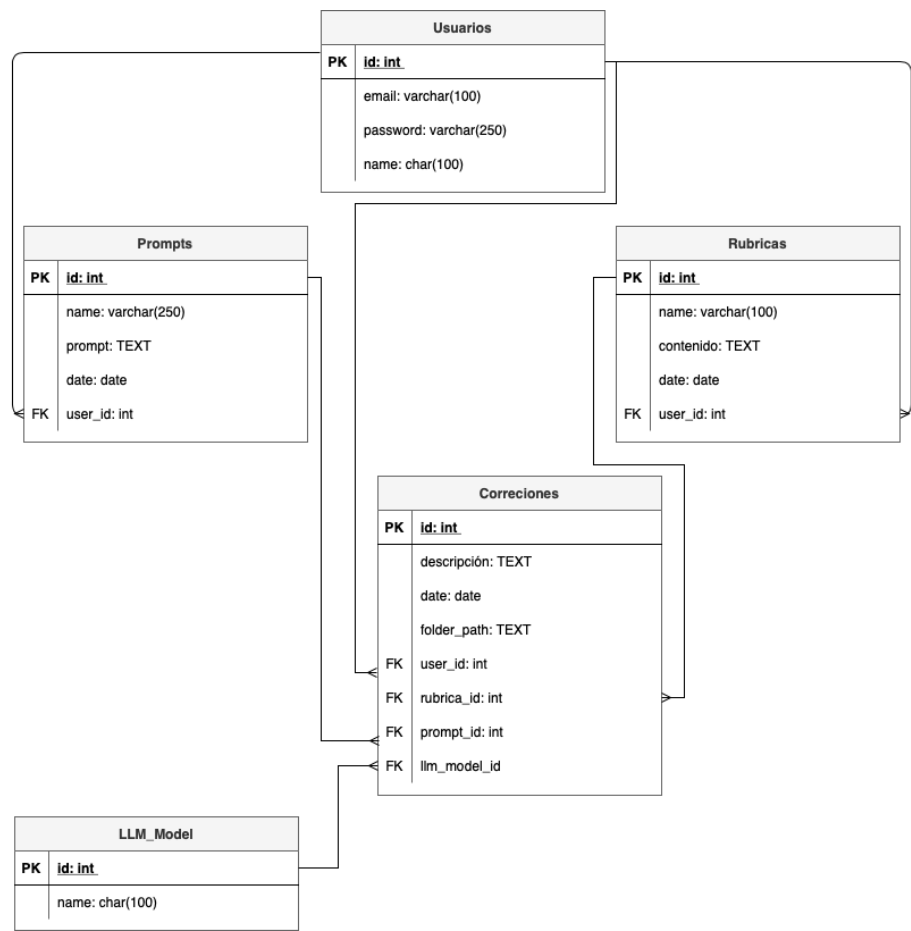


Figura C.2: Diagrama de relación

C.3. Diseño arquitectónico

C.4. Diseño procedimental

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este anexo se presenta la documentación técnica del proyecto, la cual proporciona una guía completa para cualquier programador que desee continuar con su desarrollo. Se incluyen recomendaciones sobre el entorno de desarrollo, la organización y estructura de la aplicación, los procesos de compilación, la configuración para la integración, la instalación de dependencias y las baterías de pruebas realizadas.

D.2. Estructura de directorios

Directorio Raíz (./)

El directorio raíz contiene los archivos y carpetas principales del proyecto, incluyendo configuraciones globales y documentación. Los archivos más relevantes son:

- `manage.py`: Utilidad de línea de comandos para tareas administrativas de Django, como migraciones y ejecución del servidor de desarrollo.
- `.gitignore`: Especifica los archivos y directorios excluidos del control de versiones (por ejemplo, `venv` y `.git`).
- `.pylintrc`: Configuración para la herramienta de análisis de calidad de código Pylint.

- `README.md`: Documentación inicial con instrucciones de instalación, descripción del proyecto y licencia.
- `requirements.txt`: Lista de dependencias de Python requeridas para el proyecto.

iagscore

Directorio que contiene la configuración global del proyecto Django. Aquí se encuentran los archivos principales para la configuración y enrutamiento:

- `settings.py`: Configuración del proyecto, incluyendo la base de datos, aplicaciones instaladas, claves secretas y ajustes de internacionalización.
- `urls.py`: Enrutamiento principal de URLs, que conecta las rutas de las aplicaciones con las vistas correspondientes.
- `celery.py`: Configuración para la integración de Celery, utilizado para manejar tareas asíncronas y programadas.
- `asgi.py`: Configuración para el despliegue en servidores ASGI, compatible con aplicaciones asíncronas y WebSockets.
- `wsgi.py`: Configuración para el despliegue en servidores WSGI.
- `__pycache__`: Archivos compilados de Python generados automáticamente.

accounts

Aplicación para la gestión de cuentas de usuarios (registro de usuarios, etc).

- `models.py`: Modelos personalizados para usuarios.
- `views.py`: Lógica para registro y gestión de perfiles.
- `urls.py`: Enrutamiento para URLs de registro y gestión de perfiles.
- `forms.py`: Formularios Django para la creación de nuevos usuarios.
- `admin.py`: Configuración de la interfaz de administración para los modelos de usuarios.

- `apps.py`: Configuración de la aplicación.
- `tests.py`: Pruebas unitarias para los modelos, forms, vistas y otras funcionalidades de la aplicación.
- `migrations/`: Migraciones de la base de datos.
- `static/accounts/`: Archivos estáticos específicos.
- `templates/accounts/`: Plantillas HTML para registro y gestión de perfil.

core

Aplicación que gestiona la lógica básica del proyecto como login , logout, etc.

- `models.py`: Definición de modelos de datos para la base de datos.
- `admin.py`: Configuración de la interfaz de administración de Django.
- `apps.py`: Configuración de la aplicación, incluyendo su nombre.
- `views.py`: Lógica de las vistas para procesar solicitudes HTTP.
- `urls.py`: Enrutamiento específico de la aplicación.
- `tests.py`: Pruebas unitarias para los modelos, forms, vistas y otras funcionalidades de la aplicación.
- `migrations/`: Archivos de migraciones para cambios en la base de datos.
- `static/core/`: Archivos estáticos específicos de la aplicación.
 - `static/core/docs/`: Documentación estática, como PDFs o imágenes.
- `templates/core/`: Plantillas HTML para la aplicación.
 - `templates/core/partial/`: Plantillas reutilizables para componentes modulares.

rubrics

Aplicación para gestionar rúbricas.

- `models.py`: Modelos de datos para rúbricas.
- `admin.py`: Configuración de la interfaz de administración para los modelos de rúbricas.
- `apps.py`: Configuración de la aplicación.
- `views.py`: Lógica para mostrar o gestionar rúbricas.
- `urls.py`: Enrutamiento de URLs específicas.
- `forms.py`: Formularios Django para la importación de rubricas.
- `tests.py`: Pruebas unitarias para los modelos, forms, vistas y otras funcionalidades de la aplicación.
- `migrations/`: Migraciones de la base de datos.
- `templates/rubrics/`: Plantillas HTML para mostrar o importar rúbricas.

prompts

Aplicación para gestionar prompts, entradas de texto para modelos de IA.

- `models.py`: Modelos para almacenar prompts.
- `views.py`: Lógica para gestionar prompts.
- `admin.py`: Configuración de la interfaz de administración para los modelos de prompts.
- `apps.py`: Configuración de la aplicación.
- `urls.py`: Enrutamiento de URLs.
- `forms.py`: Formularios Django para la creación de prompts.
- `tests.py`: Pruebas unitarias para los modelos, forms, vistas y otras funcionalidades de la aplicación.
- `migrations/`: Migraciones de la base de datos.
- `templates/prompts/`: Plantillas HTML para mostrar o crear prompts.

corrections

Aplicación para gestionar, configurar y ejecutar correcciones.

- `models.py`: Modelos para correcciones.
- `views.py`: Lógica para procesar correcciones.
- `admin.py`: Configuración de la interfaz de administración para los modelos de correcciones.
- `apps.py`: Configuración de la aplicación.
- `urls.py`: Enrutamiento de URLs.
- `forms.py`: Formularios Django para la gestión de correcciones.
- `tasks.py`: Definición de tareas asíncronas, gestionadas con Celery.
- `signals.py`: Configuración de señales Django para ejecutar acciones automáticas tras ciertos eventos (ej. eliminar un modelo).
- `tests.py`: Pruebas unitarias para los modelos, forms, vistas y otras funcionalidades de la aplicación.
- `migrations/`: Migraciones de la base de datos.
- `templates/corrections/`: Plantillas HTML para mostrar o configurar correcciones.

media

Directorio para archivos subidos por usuarios y almacenamiento de los resultados de la ejecución de una corrección.

- `corrections/`: Archivos relacionados con la aplicación `corrections`, organizados por IDs.

static

Carpeta para archivos estáticos globales del proyecto.

- `js/`: Archivos JavaScript para funcionalidad frontend.

tailwind

Directorio para la configuración y compilación de Tailwind CSS.

- `tailwind.config.js`: Configuración personalizada de Tailwind CSS, donde se definen temas, colores y plugins.
- `package.json`: Dependencias de Node.js y scripts para compilar Tailwind CSS.
- `static/css/`: Archivos CSS generados por Tailwind CSS.

docs

Carpeta para documentación del proyecto.

- `LaTeX/`: Documentación escrita en LaTeX.
 - `tex/`: Archivos fuente de LaTeX.
 - `img/`: Imágenes usadas en la documentación.

D.3. Manual del programador**D.4. Compilación, instalación y ejecución del proyecto****D.5. Pruebas del sistema**

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf.

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.

Bibliografía

- [1] Ken Schwaber and Jeff Sutherland. Guía de scrum: Las reglas del juego, 2020. [Internet; descargado 27-marzo-2025].