



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Plataforma Educativa para Impulsar la
Programación en Alumnos de Educación
Secundaria**

Autor: Isabella Jaimes Gelves
Tutor: Juan José Moreno Navarro

Madrid, Julio 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática

Título: Plataforma Educativa para Impulsar la Programación en Alumnos
de Educación Secundaria

Julio 2025

Autor: Isabella Jaimes Gelves

Tutor: Juan José Moreno Navarro

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería
de Software

Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Índice general

1. Introducción	4
1.1. Motivación y necesidad del proyecto	4
1.2. Objetivos	5
1.3. Estructura del documento	5
2. Estado del Arte	7
2.1. Enseñando a programar en la Educación Secundaria	7
2.1.1. Importancia de enseñar programación en secundaria	7
2.1.2. Análisis internacional	7
2.1.3. La situación de España	8
2.1.3.1. Recursos para la enseñanza de la programación	10
2.2. Estudio de páginas web similares	10
2.2.1. Project LoveLace	11
2.2.2. Codeforces	12
2.2.3. Project Euler	13
2.2.4. Acepta el Reto	14
2.2.5. Judge	15
2.3. Tecnologías empleadas	15
2.3.1. Backend	15
2.3.1.1. Django	16
2.3.1.2. Django REST Framework	16
2.3.1.3. SQLite	16
2.3.1.4. JWT (JSON Web Tokens)	16
2.3.1.5. Librerías	16
2.3.2. Frontend	17
2.3.2.1. React	17
2.3.2.2. Material-UI	17
2.3.2.3. React Router DOM	17
2.3.2.4. Styled Components	17
2.3.2.5. Librerías	17
3. Desarrollo	19
3.1. Backend	20
3.1.1. Estructura del backend	20
3.1.2. Panel de Administración	20
3.1.3. Base de Datos	22
3.1.3.1. Modelo de Datos	22
3.1.3.2. Relaciones entre tablas	22

3.1.3.3. Diagrama Entidad-Relación	23
3.1.4. API REST	23
3.1.4.1. Serializers	24
3.1.4.2. Views	24
3.1.4.3. Endpoints principales	24
3.1.5. Autenticación y autorización	24
3.1.5.1. Autenticación JWT	25
3.1.5.2. Autorización y control de acceso	25
3.1.6. Judge0	26
3.1.7. Configuración y Seguridad	26
3.2. Frontend	27
3.2.1. Tecnología y herramientas utilizadas	27
3.2.2. Estructura del frontend	27
3.2.3. Comunicación con el backend	28
3.2.4. Principales componentes y páginas	28
3.2.5. Gestión de estilos	29
3.2.6. Apariencia y usabilidad	29
3.2.7. Vista de la interfaz de usuario	30
4. Análisis de impacto	38
4.1. Impacto personal	38
4.2. Impacto Social	39
4.3. Objetivos de Desarrollo Sostenible	39
5. Resultados y conclusiones	41
5.1. Resultados	41
5.2. Trabajo futuro	42
5.3. Conclusiones	43
Bibliografía	44

Índice de figuras

2.1. Project Lovelace	11
2.2. CodeForces	12
2.3. Euler	13
2.4. Acepta el reto	14
2.5. Judge	15
3.1. Arquitectura	19
3.2. Panel de administración.	21
3.3. Diagrama base de datos	23
3.4. Flujo de autenticación con JWT	25
3.5. Integración del backend con Judge0	26
3.6. Página de bienvenida.	31
3.7. Registro	32
3.8. Inicio de sesión	33
3.9. Creación de un problema	34
3.10 Lista de problemas	35
3.11 Vista de un problema	36
3.12 Vista del envío de una solución	37

*En primer lugar, a mi familia,
por estar siempre ahí, pero especialmente a mi madre, por su apoyo incondicional, su
paciencia infinita y por ser mi mayor ejemplo de fortaleza y constancia. Gracias por no
dejarme caer nunca.*

*A Luis Miguel,
por ayudarme en todo lo que puede y más, siempre dispuesto, siempre presente. Tu
apoyo ha sido fundamental.*

*A mis amigos de la carrera,
por cada conversación, cada risa y cada momento de apoyo. Gracias por estar cuando
más os he necesitado y por hacer este camino mucho más llevadero.*

*A Juan José Moreno,
por su dedicación, su orientación y sus valiosos consejos a lo largo de este trabajo.*

*A mi Moka,
por su compañía silenciosa y su alegría constante, que me han acompañado en cada
paso de este camino.*

*Y a mi Alejandro,
por su amor constante, por creer en mí incluso cuando yo no lo hacía y por recordarme
quién soy cuando más lo necesito.*

Resumen

Este Trabajo de Fin de Grado recoge el desarrollo e investigación realizados en torno a **CodeChallenges**, una plataforma web educativa orientada a estudiantes de Educación Secundaria para el aprendizaje de la programación, integrando además contenidos de distintas asignaturas.

La plataforma permite a los docentes crear y subir problemas de programación personalizados, adaptados a las necesidades de sus alumnos.

A diferencia de otras plataformas similares, CodeChallenges se caracteriza por su flexibilidad y dinamismo, ya que no limita la creación de contenido a los desarrolladores de la plataforma. Cualquier profesor puede generar y compartir sus propios problemas, lo que fomenta una comunidad activa y en constante crecimiento.

Palabras Clave: Plataforma educativa, Evaluación automática, Programación, Educación secundaria.

Abstract

This Final Degree Project presents the development and research carried out around **CodeChallenges**, an educational web platform aimed at secondary education students for learning programming, while also integrating content from different subjects.

The platform allows teachers to create and upload personalized programming problems, tailored to the needs of their students.

Unlike other similar platforms, CodeChallenges stands out for its flexibility and dynamism, as it does not restrict content creation to the platform's developers. Any teacher can generate and share their own problems, fostering an active and constantly growing community.

Keywords: Educational platform, Automatic evaluation, Programming, Secondary education.

Capítulo 1

Introducción

1.1. Motivación y necesidad del proyecto

En los últimos años, la programación ha adquirido un papel fundamental en el ámbito educativo, especialmente a raíz de la aprobación de la Ley Orgánica 3/2020 (LOMLOE) [1], que promueve la inclusión de contenidos relacionados con la tecnología, la computación y el pensamiento computacional en los planes de estudio de la educación obligatoria.

Siguiendo esta línea, el Decreto 65/2022 [2], de 20 de julio, del Consejo de Gobierno de la Comunidad de Madrid, establece que el alumnado debe recibir formación en *Tecnología y Digitalización*, que incluye la enseñanza de la programación, en al menos dos cursos de Educación Secundaria Obligatoria.

No obstante, la aplicación de esta normativa presenta retos significativos para los centros educativos, y en particular para el profesorado. Muchos docentes no disponen de los recursos académicos adecuados o de formación especializada que les permitan impartir contenidos de programación de forma eficaz. Al ser una materia nueva, hay una carencia de material educativo al respecto adaptado a la realidad del alumnado español. El profesorado necesita acudir a algunas plataformas existentes que ofrecen ejercicios para realizar en el aula,

Sin embargo, las plataformas actuales suelen centrarse en ejercicios cerrados y homogéneos, sin una conexión directa con otras materias del currículo. Esta falta de integración impide que los estudiantes comprendan la aplicabilidad real de lo que están aprendiendo, lo que repercute negativamente en su motivación y en la percepción de la programación como una herramienta útil.

Ante esta situación, surge la necesidad de desarrollar una plataforma educativa que no solo enseñe programación, sino que lo haga mediante la aplicación práctica de esta en otras áreas del conocimiento. Este enfoque interdisciplinar permite que los estudiantes comprendan la programación como una herramienta para resolver problemas reales, modelar situaciones y reforzar contenidos de asignaturas como matemáticas, física, biología o lengua.

Este proyecto tiene como objetivo dar respuesta a esa necesidad mediante el desarrollo de una página web que permita al profesorado diseñar, gestionar y compartir ejercicios de programación contextualizados, adaptados a distintos niveles de dificultad.

Introducción

tad, curso y materias. A su vez, el alumnado podrá acceder a estos retos, resolverlos de forma interactiva y aprender no solo a programar, sino a aplicar sus conocimientos sobre otras asignaturas de forma significativa, lo que, adicionalmente, reforzará esas materias.

En resumen, la motivación principal de este trabajo es aportar una solución educativa innovadora, flexible y útil para fomentar el aprendizaje significativo de la programación en secundaria, apoyando tanto al profesorado en su labor docente como al alumnado en su desarrollo competencial.

1.2. Objetivos

El objetivo principal de este trabajo es desarrollar una plataforma web educativa orientada a estudiantes y docentes de Educación Secundaria Obligatoria. La finalidad de esta plataforma es fomentar el aprendizaje de la programación mediante la resolución de problemas contextualizados en distintas asignaturas del currículo escolar. Además, se busca proporcionar un entorno accesible, atractivo e interactivo, que facilite tanto la enseñanza como la práctica autónoma, adaptándose a distintos niveles y necesidades pedagógicas.

- Desarrollar una plataforma web que permita a los estudiantes de secundaria practicar el desarrollo de programas aplicados a problemas reales de diversas asignaturas del plan de estudio.
- Diseñar una interfaz intuitiva y accesible para que la plataforma sea fácil de usar tanto para docentes como para estudiantes.
- Facilitar la creación y gestión de problemas por parte del profesorado.
- Implementar un sistema de clasificación y etiquetado de problemas basado en criterios como materia, nivel de dificultad y tiempo estimado de resolución.
- Integrar un sistema de evaluación automática que permita validar los programas entregados por los estudiantes, verificando su corrección y eficiencia.

1.3. Estructura del documento

El resto de la memoria se estructura de la siguiente forma:

- El capítulo 2, titulado “Estado del arte”, presenta un resumen de la situación actual de la programación en la educación secundaria, destacando su creciente relevancia curricular. Además, se analizan distintas alternativas y tecnologías existentes, subrayando aquellas que se han considerado más apropiadas para el desarrollo del proyecto.
- El capítulo 3, titulado “Desarrollo”, describe el proceso técnico llevado a cabo para construir la plataforma web. Se explica cómo se ha implementado el *backend* con Django, definiendo los modelos y exponiendo los datos mediante una API REST, la integración de Judge0 y cómo se ha desarrollado el *frontend* con React, construyendo una interfaz atractiva, intuitiva y dinámica para usuarios y docentes.

1.3. Estructura del documento

- El capítulo 4, titulado “Impacto del trabajo”, presenta un análisis del alcance y la repercusión del proyecto en distintos ámbitos, como el educativo, social y personal. Se valoran los beneficios que puede aportar la plataforma tanto para docentes como para estudiantes, así como su alineación con objetivos sociales actuales como la digitalización y la igualdad de oportunidades en el acceso a la educación tecnológica.

Capítulo 2

Estado del Arte

2.1. Enseñando a programar en la Educación Secundaria

En los últimos años, la programación ha experimentado una notable evolución, dejando de ser una disciplina exclusiva del ámbito universitario o profesional [3]. Actualmente, se integra cada vez más en las etapas educativas tempranas. Esta transformación la ha consolidado como un pilar fundamental en la formación de competencias digitales, el desarrollo del **pensamiento computacional** y el fomento de habilidades **STEM** (Ciencia, Tecnología, Ingeniería y Matemáticas) en los jóvenes.

Muchos jóvenes, interesados por la informática y la programación, animados por sus familias, han hecho que florecieran clubs de programación, actividades como las de Code.org, hackatones, campamentos, extraescolares, etc.

2.1.1. Importancia de enseñar programación en secundaria

La integración de la programación en esta etapa educativa ofrece beneficios clave [4, 5] como:

- La programación no solo prepara a los estudiantes para carreras tecnológicas, sino que también desarrolla habilidades transversales esenciales como la resolución de problemas, el pensamiento lógico y la creatividad. [6]
- Aprender a programar puede abrir la puerta a estudios y profesiones STEM, campos con una alta demanda laboral, pero también es hoy un importante elemento de otras profesiones, como las del ámbito de salud o la sociología. [7]
- Potencia la expresión creativa y la capacidad de enfrentar retos complejos, habilidades valiosas en cualquier ámbito profesional o personal. [8]

2.1.2. Análisis internacional

En Europa, cada vez más países han apostado por la programación y el pensamiento computacional en sus currículos escolares. En 2014, un estudio realizado en 20 Ministerios de Educación de la Unión Europea reveló que en 12 de ellos la programación ya estaba integrada en su plan de estudios, y otros siete países tenían planes de introducirla a corto plazo [9].

2.1. Enseñando a programar en la Educación Secundaria

Países como Estonia, Reino Unido y Finlandia han sido pioneros en esta iniciativa. Por ejemplo, Estonia, con su programa ProgeTiiger, enseña programación desde la educación primaria. El Reino Unido la hizo obligatoria en primaria y secundaria en 2014, mientras que Finlandia la incorporó en primaria a partir de 2016. [9, 10]

Fuera del contexto europeo, naciones como Singapur y China también han implementado reformas educativas significativas. En Singapur, los estudiantes comienzan a desarrollar el pensamiento computacional desde la educación infantil y, en primaria, empiezan a aprender Scratch. Por su parte, el gobierno de China incorporó en 2016 un plan educativo que inicia a los 6 años y progresa en complejidad a medida que los estudiantes avanzan en edad. [10]

Corea del Sur, tras una reforma en 2015, ha integrado la educación en software y programación en todos los niveles educativos. Allí, la programación se imparte como una asignatura específica, con un fuerte enfoque en el desarrollo del pensamiento computacional y la creatividad. Además, se han puesto en marcha programas de formación docente para asegurar que el profesorado cuente con las competencias necesarias para impartir estos contenidos, una estrategia también adoptada en Finlandia. [11]

2.1.3. La situación de España

En este contexto, España ha dado un paso decisivo con la entrada en vigor de la LOMLOE (Ley Orgánica 3/2020), que impulsa una educación más competencial e interdisciplinar. Uno de los cambios más relevantes es la introducción de la asignatura **Tecnología y Digitalización**. Esta asignatura está diseñada para que el alumnado desarrolle habilidades prácticas y teóricas en áreas clave como la programación, el pensamiento computacional, la resolución de problemas tecnológicos, el diseño digital o el uso ético de la tecnología.

A continuación, se resumen algunas de las competencias específicas de esta materia, según el anexo del currículo:

Nº	Competencia específica	Descripción resumida
CE1	Identificación de necesidades tecnológicas	Analizar situaciones cotidianas para proponer soluciones tecnológicas viables.
CE2	Pensamiento computacional	Aplicar principios de programación para resolver problemas mediante algoritmos.
CE3	Desarrollo de soluciones digitales	Diseñar y desarrollar programas que den respuesta a desafíos concretos.
CE4	Evaluación de impactos de la tecnología	Reflexionar críticamente sobre las implicaciones éticas, sociales y medioambientales del uso de la tecnología.
CE5	Diseño de proyectos tecnológicos	Planificar y ejecutar proyectos tecnológicos colaborativos, usando recursos digitales y físicos.
CE6	Ciberseguridad y uso responsable	Conocer principios básicos de ciberseguridad y fomentar una ciudadanía digital segura.

Cuadro 2.1: Competencias específicas [12]

Este cambio representa un punto de inflexión: la programación, que antes era una materia optativa que escogían alumnos ya interesados, ahora forma parte del currículo general. Esto significa que abarca a estudiantes con perfiles muy diversos, lo que plantea un claro reto motivacional. La programación deja de ser un espacio exclusivo para quienes ya sentían afinidad tecnológica para convertirse en un contenido común que debe conectar con todo el alumnado, incluyendo a aquellos más orientados hacia las humanidades o las artes o sin habilidades computacionales. Como indican varios estudios [13], los "nativos digitales" no lo son tanto. Por lo tanto, la necesidad de estrategias motivadoras y pedagógicamente eficaces es ahora una prioridad.

También plantea un reto organizativo: tomemos como referencia un centro educativo tipo, de línea 4 (esto es, cuatro grupos de cada año). Este centro pasaría de 1 asignatura optativa (con una media de 20 estudiantes) a 8 grupos (4 años por cada una de las nuevas asignaturas, completas, que pueden suponer unos 300 estudiantes). Supone más profesorado y más aulas informáticas, pero también una gestión compleja, pues los niveles entre los alumnos serán diferentes y la evaluación de ejercicios puede suponer un esfuerzo considerable.

Una de las vías más prometedoras para integrar la programación en la educación es relacionarla con problemas cercanos a los intereses del alumnado, conectándola con otras materias. Esta aproximación transversal permite un aprendizaje basado en competencias, favoreciendo la conexión entre saberes y destacando la aplicabilidad del conocimiento.

Además, es crucial incorporar herramientas atractivas y accesibles, como entornos de programación por bloques, dispositivos físicos como Arduino o MicroBit, o plataformas con una estética y lógica acordes con la visión adolescente. Esta combinación de problemas contextualizados y herramientas motivadoras se alinea con metodologías activas como el aprendizaje basado en proyectos (ABP) o el enfoque STEAM. Esto no solo facilita la adquisición de competencias digitales, sino que también fomenta el desarrollo de la creatividad, el trabajo en equipo y la autonomía.

La herramienta puede también liberar de trabajo a los docentes, si incluye capacidades para recibir trabajos y evaluarlos automáticamente, reportando los resultados al profesor,

Este contexto refuerza la necesidad de contar con herramientas educativas que resulten realmente motivadoras para el alumnado. En particular, se identifican dos estrategias clave que guían el desarrollo de este proyecto:

1. **Aplicación transversal de la programación:** en lugar de presentar ejercicios abstractos o desvinculados del resto del currículo, se propone que los estudiantes resuelvan problemas contextualizados en materias que ya les son familiares (como Matemáticas, Biología o Física). Este enfoque favorece un aprendizaje más significativo, fomenta la interdisciplinariedad y refuerza la adquisición de competencias clave, como el pensamiento lógico, la resolución de problemas o la autonomía en el aprendizaje.
2. **Diseño visual y comunicativo adaptado al alumnado adolescente:** la plataforma se ha concebido pensando en su público objetivo. Para ello, se ha optado por una estética visual inspirada en el cómic, un lenguaje cercano y dinámico, y una interfaz que evita los formalismos excesivos característicos de otras plataformas educativas. El objetivo es generar una experiencia más atractiva,

accesible y afín a ellos.

Ambas decisiones buscan no solo facilitar la enseñanza de la programación, sino también conectar con los intereses y formas de aprender del alumnado, incrementando así su implicación y motivación en el proceso educativo.

Aunque el proyecto está pensado principalmente para su integración escolar en secundaria, puede ser una ayuda para otras personas que quieran adentrarse por sí mismos en el aprendizaje de la programación. En particular, aquellos jóvenes interesados en aprender a programar y divertirse con ello y que sus familias no puedan permitirse abonar los costes de actividades adicionales o extraescolares (cursos, clubes de programación, campamentos, etc.) pueden encontrar en CodeChallenges una herramienta libre para ejercitar su afición.

2.1.3.1. Recursos para la enseñanza de la programación

Existen múltiples herramientas y plataformas adaptadas a la educación secundaria que facilitan la enseñanza y el aprendizaje de la programación en diferentes facetas. Se mencionan algunos:

- **Lightbot:** Juego educativo que enseña conceptos básicos de programación y pensamiento computacional a través de la resolución de puzzles [14].
- **Scratch y Blockly:** Lenguajes visuales ideales para introducir conceptos básicos de programación de forma lúdica y accesible [15, 14].
- **CS First (Google):** Plataforma gratuita con vídeos, tutoriales y planes de clase que permite gestionar grupos y personalizar actividades, ideal para docentes sin experiencia previa [15].
- **Esobots:** Proyecto basado en Arduino que integra robótica y programación, alineado con el currículo de Tecnología y Digitalización en secundaria [15, 14].
- **ROBOTIX C360:** Plataforma con más de 500 recursos curriculares, que facilita la integración de la programación y la robótica en el aula, optimizando la preparación y evaluación docente [15, 14].
- **Code.org y Codecademy:** Plataformas con tutoriales interactivos para aprender desde los fundamentos hasta lenguajes como Python o JavaScript, adaptados a diferentes niveles [15, 14].
- **Banana Coders:** Plataforma española que fomenta la enseñanza de la programación en edades tempranas con un enfoque lúdico y colaborativo [14].

2.2. Estudio de páginas web similares

En la actualidad, existen diversas plataformas de evaluación automática que ofrecen funcionalidades similares a las que se plantean en esta propuesta. No obstante, muchas de estas herramientas presentan un número limitado de ejercicios o no están específicamente orientadas al nivel educativo de los estudiantes de secundaria.

En las siguientes secciones se analizarán las ventajas e inconvenientes de algunas de estas plataformas, con el objetivo de identificar elementos clave para el diseño y desarrollo de la solución propuesta.

2.2.1. Project LoveLace

Project Lovelace [16] es una plataforma educativa que ofrece ejercicios de programación sobre física, química, biología o matemáticas.

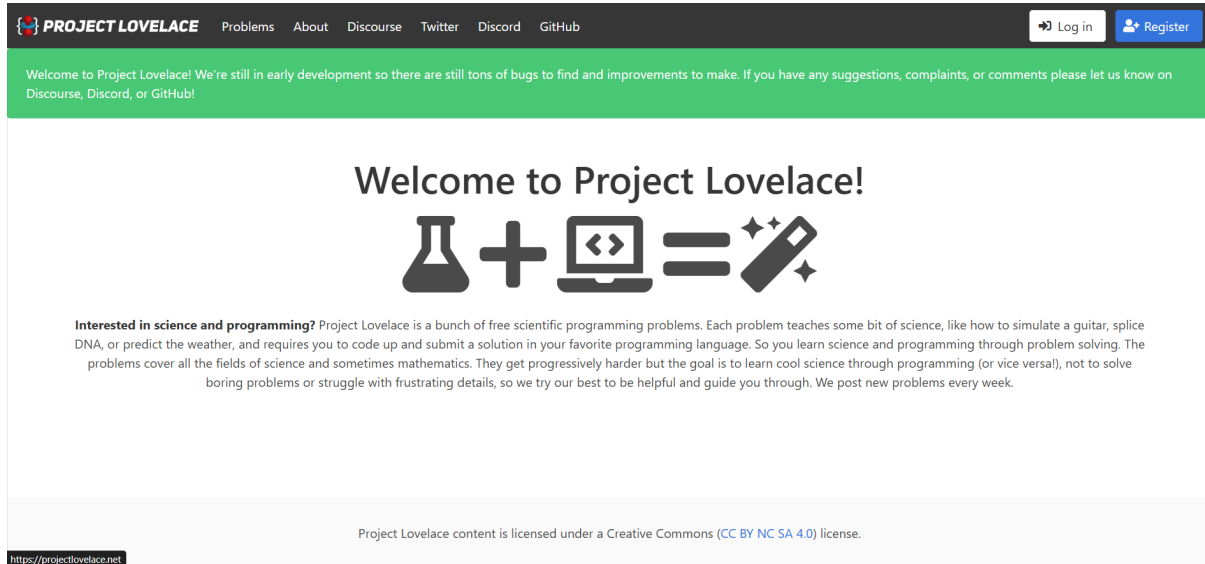


Figura 2.1: Project Lovelace

Ventajas:

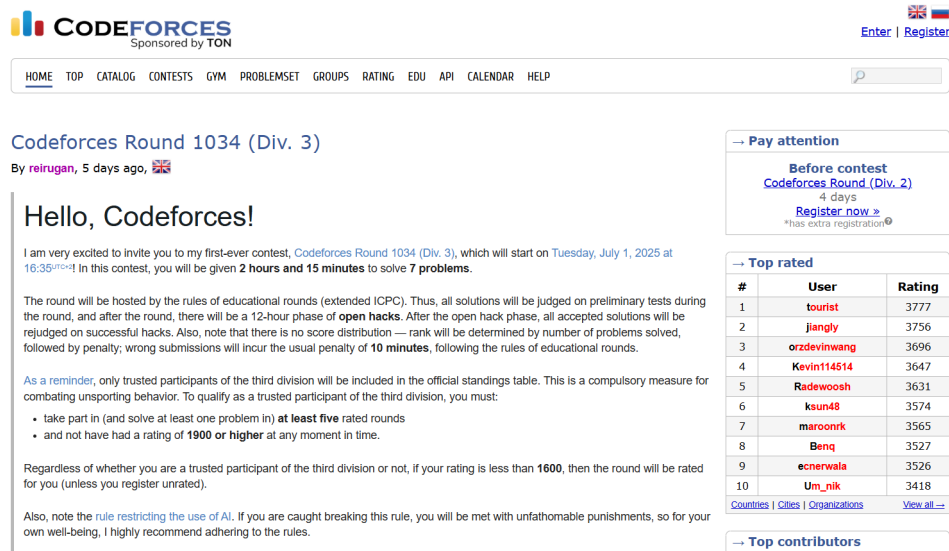
- Enfoque divulgativo y multidisciplinar.
- Interfaz sencilla y accesible.
- No requiere registro para comenzar a practicar.

Inconvenientes:

- Número muy limitado de problemas (menos de 30).
- Falta de adaptación a niveles básicos, lo que puede dificultar su uso en secundaria.
- No permite que los docentes personalicen o suban nuevos ejercicios.
- Lleva sin actualizarse varios años.

2.2.2. Codeforces

Codeforces [17] es una plataforma de programación competitiva orientada a estudiantes avanzados y profesionales. Está centrada en concursos y problemas de alta dificultad.



CODEFORCES
Sponsored by TON

Enter | Register

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

Codeforces Round 1034 (Div. 3)

By [reirugan](#), 5 days ago,

Hello, Codeforces!

I am very excited to invite you to my first-ever contest, [Codeforces Round 1034 \(Div. 3\)](#), which will start on **Tuesday, July 1, 2025 at 16:35 UTC+1**. In this contest, you will be given **2 hours and 15 minutes** to solve **7 problems**.

The round will be hosted by the rules of educational rounds (extended ICPC). Thus, all solutions will be judged on preliminary tests during the round, and after the round, there will be a 12-hour phase of **open hacks**. After the open hack phase, all accepted solutions will be rejudged on successful hacks. Also, note that there is no score distribution — rank will be determined by number of problems solved, followed by penalty; wrong submissions will incur the usual penalty of **10 minutes**, following the rules of educational rounds.

As a reminder, only trusted participants of the third division will be included in the official standings table. This is a compulsory measure for combating unsporting behavior. To qualify as a trusted participant of the third division, you must:

- take part in (and solve at least one problem in) **at least five** rated rounds
- and not have had a rating of **1900 or higher** at any moment in time.

Regardless of whether you are a trusted participant of the third division or not, if your rating is less than **1600**, then the round will be rated for you (unless you register unrated).

Also, note the [rule restricting the use of AI](#). If you are caught breaking this rule, you will be met with unfathomable punishments, so for your own well-being, I highly recommend adhering to the rules.

→ Pay attention

Before contest
[Codeforces Round \(Div. 2\)](#)
4 days
[Register now >](#)
*has extra registration

→ Top rated

#	User	Rating
1	tourist	3777
2	jiangly	3756
3	orzdevinwang	3696
4	Kevin114514	3647
5	Radewoosh	3631
6	ksun48	3574
7	maroonrk	3565
8	Benq	3527
9	ecnerwala	3526
10	Um_nik	3418

[Countries](#) | [Cities](#) | [Organizations](#) | [View all](#)

→ Top contributors

Figura 2.2: CodeForces

Ventajas:

- Amplio repositorio de problemas clasificados por dificultad.
- Sistema robusto de envío y evaluación automática.
- Comunidad activa y recursos para práctica avanzada.

Inconvenientes:

- Alta complejidad: los problemas no están orientados a niveles iniciales ni adaptados a secundaria.
- Poco enfoque educativo o interdisciplinar.
- Interfaz y entorno algo abrumadores para usuarios novatos.

2.2.3. Project Euler

Project Euler [18] combina matemáticas y programación, con una colección de retos algorítmicos que requieren razonamiento lógico.

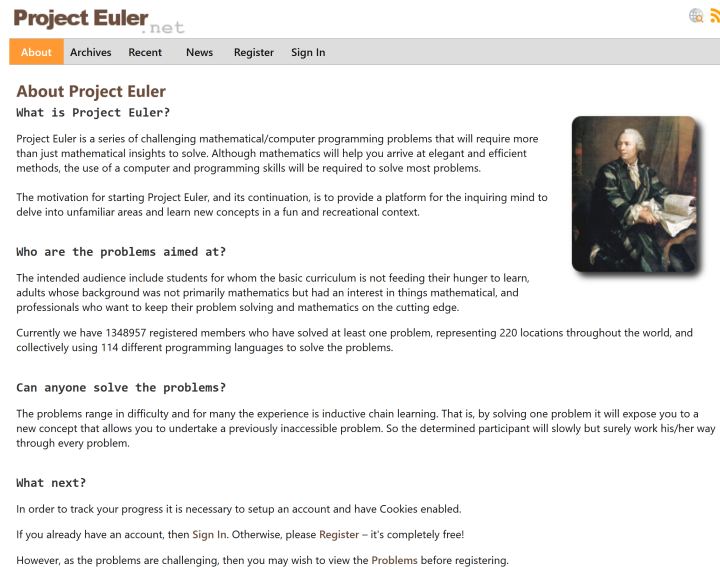


Figura 2.3: Euler

Ventajas:

- Excelente para fomentar el pensamiento computacional.
- Relación clara entre programación y matemáticas.
- Muy valorada en comunidades técnicas.

Inconvenientes:

- No dispone de un sistema de evaluación automática tradicional (el usuario introduce una respuesta numérica).
- Los problemas suelen ser complejos y poco accesibles para niveles de iniciación.
- No existe opción para subir nuevos ejercicios o personalizarlos.

2.2.4. Acepta el Reto

Acepta el reto [19] es una plataforma desarrollada en España, muy utilizada en contextos universitarios, orientada a la programación académica. Incluye problemas clásicos con evaluación automática.



Figura 2.4: Acepta el reto

Ventajas:

- Amplia base de problemas.
- Plataforma estable y mantenida por instituciones académicas.
- Formato estandarizado para concursos y prácticas.

Inconvenientes:

- Enfoque poco flexible: no está orientado a docentes de secundaria.
- Interfaz algo técnica y menos amigable para alumnado joven.
- No ofrece herramientas para adaptar el contenido a materias escolares.

2.2.5. Jutge

Jutge.org [20] es una plataforma de evaluación automática que permite practicar problemas de programación en múltiples lenguajes.

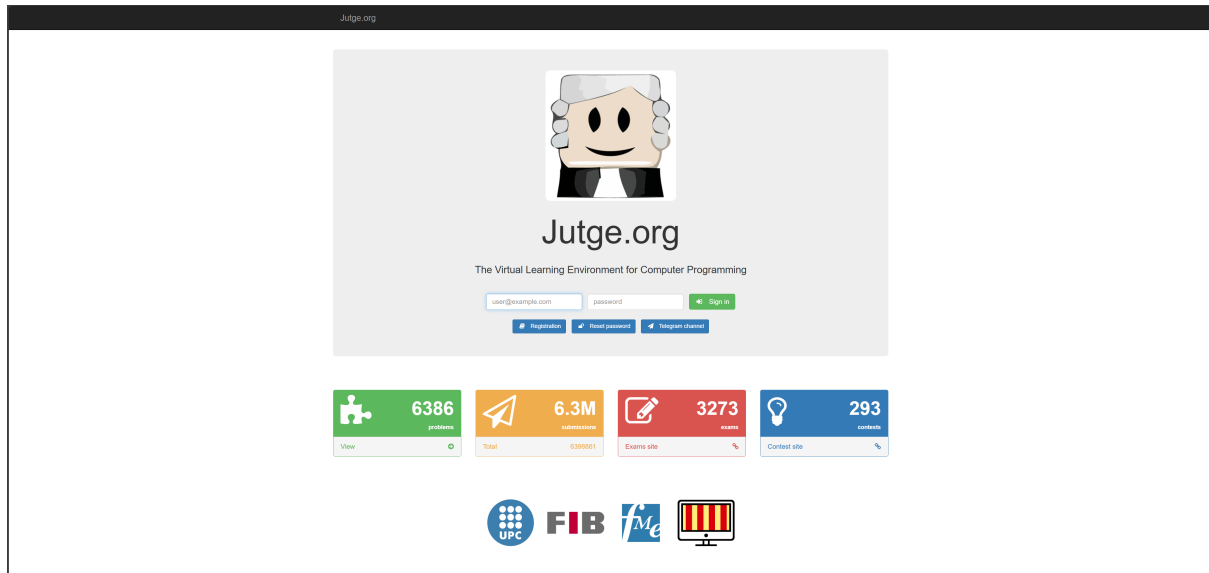


Figura 2.5: Jutge

Ventajas:

- Permite la práctica en distintos lenguajes de programación.
- Evaluación automática eficiente.
- Recursos variados y con cierta estructura educativa.

Inconvenientes:

- Interfaz menos accesible para estudiantes jóvenes.
- No está pensada para su integración curricular directa.
- Limitada en cuanto a personalización por parte de los docentes.

2.3. Tecnologías empleadas

En este apartado se describen las tecnologías utilizadas a lo largo del desarrollo del proyecto, justificando su elección en función de los requisitos del sistema y destacando sus principales características.

2.3.1. Backend

El *backend* es la parte de la aplicación que se ejecuta en el servidor y que no es visible para el usuario final. Su función principal es gestionar la lógica de negocio, procesar

las solicitudes del frontend, acceder a la base de datos y garantizar el correcto funcionamiento de la aplicación mediante el manejo de la información y la comunicación entre los distintos componentes del sistema.

2.3.1.1. Django

Para el desarrollo del backend se ha optado por el framework Django [21], debido a su estructura clara y a su capacidad para crear aplicaciones web robustas de forma rápida y segura. Basado en el patrón Model-View-Template (MVT), Django facilita la organización del código y separa eficazmente la lógica de negocio de la presentación.

Incluye un ORM (Object-Relational Mapper) integrado que simplifica la interacción con la base de datos, así como herramientas ya incorporadas para la gestión de usuarios, autenticación, permisos y administración del sitio. Su amplia comunidad, documentación extensa y madurez como tecnología lo convierten en una opción fiable para proyectos educativos como este.

2.3.1.2. Django REST Framework

Django REST Framework (DRF) [22] es una extensión de Django diseñada para simplificar el desarrollo de APIs RESTful. Esta herramienta incorpora buenas prácticas y proporciona utilidades como la serialización de datos, gestión de permisos, autenticación y creación de endpoints personalizados. Gracias a su integración con Django, permite exponer de forma sencilla y segura los datos del backend, facilitando la comunicación con el frontend mediante peticiones HTTP.

2.3.1.3. SQLite

SQLite ofrece una solución de base de datos ligera, autónoma y sin necesidad de configuración, que almacena todos los datos en un único archivo en disco. Esta característica la convierte en una opción ideal para entornos de desarrollo y prototipado, al proporcionar una configuración sencilla y un rendimiento eficiente sin requerir un servidor de base de datos dedicado.

2.3.1.4. JWT (JSON Web Tokens)

JWT [23] permite implementar de forma segura un sistema de autenticación sin estado, lo que facilita la transmisión de información de autenticación entre el cliente y el servidor. Gracias a este enfoque, no es necesario mantener sesiones activas en el servidor, lo que simplifica la escalabilidad de la aplicación y mejora el rendimiento.

2.3.1.5. Librerías

En Python, el uso de librerías es muy común debido a *pip*, su gestor de paquetes. Esto facilita enormemente su empleo, ya que las dependencias no representan ningún inconveniente. Entre ellas destacan:

- *django-cors-headers*: Gestiona las políticas CORS (Cross-Origin Resource Sharing) para permitir la comunicación entre el frontend y el backend.
- *python-dotenv*: Se utiliza para administrar de forma segura las variables de entorno y los ajustes confidenciales.

- *djangorestframework-simplejwt*: Para la implementación de autenticación JWT integrada con Django REST Framework.
- *PyJWT*: Se encarga de la creación y verificación de tokens JWT en el backend.

2.3.2. Frontend

El *frontend* es la parte de la aplicación que se ejecuta en el navegador web y con la que interactúan los usuarios. Su función principal es presentar la información de forma visual e intuitiva, gestionar las acciones del usuario (como clics o formularios) y comunicarse con el backend para la obtención o el envío de datos.

2.3.2.1. React

React [24] ha sido seleccionado como la biblioteca principal para el desarrollo del frontend debido a su enfoque basado en componentes, que facilita la construcción de interfaces modulares, reutilizables y mantenibles. Su sistema de gestión de estado y el uso del Virtual DOM permiten optimizar el rendimiento de la aplicación, lo que lo convierte en una solución robusta y eficiente.

2.3.2.2. Material-UI

Material-UI aplica los principios de Material Design propuestos por Google. Proporciona elementos predefinidos con un diseño consistente, accesible y adaptable, lo que facilita la construcción de interfaces modernas y usables.

2.3.2.3. React Router DOM

React Router DOM proporciona un sistema de enrutamiento declarativo y eficiente, que facilita la creación de aplicaciones de una sola página (SPA). Permite gestionar la navegación entre vistas sin recargar la página, manteniendo la experiencia de usuario y el historial del navegador de forma coherente.

2.3.2.4. Styled Components

Styled Components permite definir componentes con estilos encapsulados directamente en JavaScript, lo que facilita la organización del código y favorece la implementación de temas dinámicos de forma eficiente y escalable.

2.3.2.5. Librerías

El uso de librerías en JavaScript está ampliamente extendido gracias al gestor de paquetes *npm* (Node Package Manager), que facilita su integración y gestión sin complicaciones relacionadas con las dependencias. Entre las más destacadas se encuentran:

- *React Hook Form*: se utiliza para la administración eficaz de los formularios con validación integrada.
- *Axios*: Para realizar solicitudes HTTP de manera sencilla y coherente.
- *React Toastify*: Para mostrar notificaciones elegantes y configurables.

2.3. Tecnologías empleadas

- *JWT Decode*: Para el manejo seguro de tokens *JWT* (JSON Web Tokens).
- *Yup*: Para implementar validaciones de esquemas robustas y extensibles.

Capítulo 3

Desarrollo

En este capítulo se expone el proceso seguido para el desarrollo del proyecto.

La fase de requisitos ha sido descrita anteriormente, en la medida de que se han indicado las características que debe tener el proyecto.

Para la arquitectura, se ha elegido el estilo REST o de *transferencia de estado representacional* (en inglés *representational state transfer*) para diseñar sistemas hipermedia distribuidos. El proyecto sigue una arquitectura de aplicación web distribuida, donde el *frontend* y el *backend* operan como componentes independientes que se comunican e intercambian manera segura (APIs REST). Esto permite la separación clara de responsabilidades.

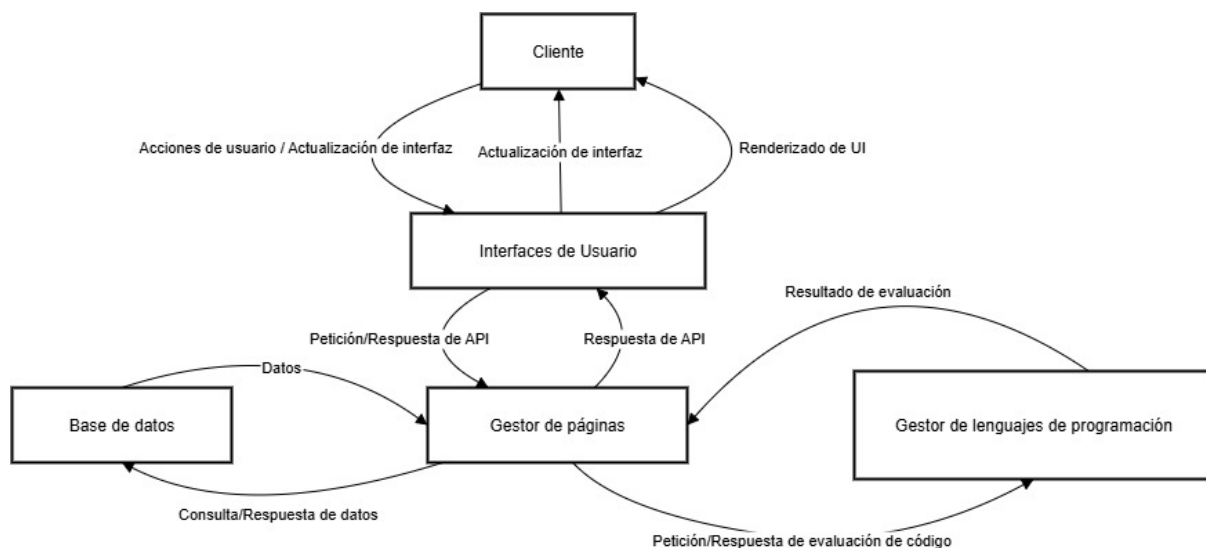


Figura 3.1: Arquitectura

Las secciones siguientes describen sus elementos principales, a la vez que se dan detalles de su implementación.

3.1. Backend

En esta sección se explica todo lo relacionado con la lógica del servidor, la gestión de datos y la comunicación con servicios externos. El backend de este proyecto está desarrollado con *Django* (Python) y expone una *API RESTful* que permite la interacción entre el frontend y la base de datos, así como la integración con el servicio externo *Judge0* para la ejecución y validación de código.

3.1.1. Estructura del backend

La estructura del backend se organiza en diferentes carpetas, cada una con una función específica:

- **accounts:** contiene la configuración relacionada con la gestión y visualización de los usuarios. Aquí se encuentran archivos clave como `models.py` (definición de usuarios), `serializers.py`, `views.py`, y `permissions.py` (gestión de permisos), así como archivos para la administración, rutas y pruebas.
- **api:** define la lógica principal de la aplicación, gestionando los problemas y soluciones. Incluye archivos para modelos, vistas, serializadores, rutas y administración.
- **backend:** agrupa la configuración global del proyecto, incluyendo los archivos de configuración (`settings.py`, `urls.py`, `wsgi.py`, `asgi.py`) y los requisitos del entorno (`requirements.txt`).
- **manage.py:** archivo principal para la gestión y administración del proyecto Django.

Esta estructura permite separar claramente la gestión de usuarios, la lógica de la aplicación y la configuración global, facilitando el mantenimiento y la escalabilidad del proyecto.

3.1.2. Panel de Administración

Para la gestión interna de usuarios, problemas y otros datos relevantes, se ha utilizado el panel de administración predeterminado de Django. Este panel proporciona una interfaz web robusta y segura que permite a los administradores del sistema crear, modificar y eliminar registros de manera sencilla, sin necesidad de desarrollar una herramienta personalizada desde cero.

El uso del panel de Django facilita la administración del proyecto y acelera el desarrollo, ya que ofrece funcionalidades avanzadas como la gestión de permisos, la búsqueda y la edición en línea de los modelos definidos en la aplicación.

Start typing to filter...

- ACCOUNTS
 - Users [+ Add](#)
- API
 - Problems [+ Add](#)
 - Solutions [+ Add](#)
 - Test cases [+ Add](#)

Select problem to change

Action: 0 of 11 selected

- PROBLEM
- Determinar si un número es capicúa
- Número con cifras en orden inverso
- Obtener la última cifra de un número
- Obtener la primera cifra de un número
- Concatenar dos números
- Convertir una lista de dígitos en un número
- Contar el número de cifras de un número
- Comprobar si un dígito está en un número
- Suma de las cifras de un número
- Obtener la lista de cifras de un número
- Número más alto con dos dígitos

11 problems

[ADD PROBLEM](#)

Figura 3.2: Panel de administración.

3.1.3. Base de Datos

La base de datos es un componente fundamental del sistema, ya que almacena toda la información relativa a los usuarios, problemas, soluciones y casos de prueba. En este proyecto se utiliza *SQLite3* durante el desarrollo.

En Django, la integración con la base de datos se realiza principalmente a través de los modelos. Estos son clases de Python que representan directamente las tablas y sus relaciones en la base de datos. Cuando creas o modificas un modelo, Django utiliza migraciones para aplicar esos cambios a la base de datos real de forma automática y controlada.

La interacción con los datos se realiza mediante el **ORM** (Object-Relational Mapper) de Django. Esto te permite crear, consultar y modificar registros usando código Python, sin necesidad de escribir SQL directamente.

Las vistas y los serializers se encargan de recibir las peticiones de los usuarios, validar y transformar los datos, y luego usar el ORM para guardar o recuperar la información en la base de datos. Toda esta configuración se gestiona fácilmente desde los archivos del proyecto y puede administrarse visualmente desde el panel de administración de Django, lo que simplifica enormemente el desarrollo y mantenimiento del backend.

3.1.3.1. Modelo de Datos

El modelo de datos está diseñado para sustentar una plataforma de problemas de programación. Este modelo permite gestionar eficientemente a los usuarios (profesores y alumnos), los problemas, las soluciones y los casos de prueba.

Tablas del Datos:

- **User:** Esta tabla almacena toda la información de los usuarios de la plataforma, incluyendo su rol (profesor, alumno o administrador).
- **Problema:** Aquí se guardan los enunciados de los problemas, junto con atributos como su nivel, dificultad y el autor que lo creó.
- **TestCase:** Contiene los casos de prueba que se asocian a cada problema. Cada caso de prueba incluye la entrada y la salida esperada para verificar las soluciones.
- **Solución:** Esta tabla registra las soluciones que los alumnos envían para cada problema específico.

3.1.3.2. Relaciones entre tablas

- Un **usuario** (profesor) puede crear varios **problemas**.
- Un **problemas** debe tener al menos un **test case**.
- Un **usuario** (alumno) puede enviar varias **soluciones**.
- un **problemas** puede recibir varias **soluciones** de diferentes alumnos.

3.1.3.3. Diagrama Entidad-Relación

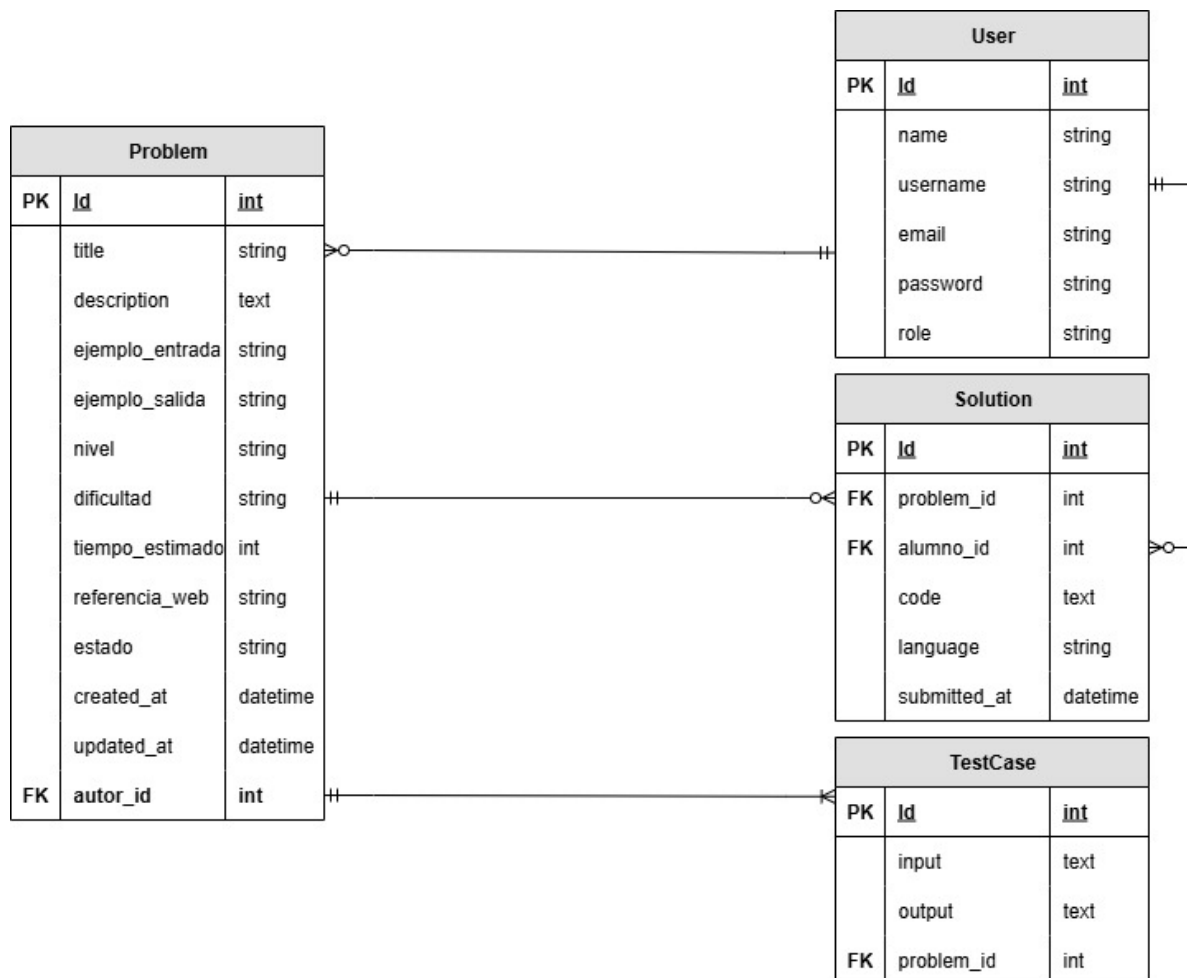


Figura 3.3: Diagrama base de datos

3.1.4. API REST

La API REST actúa como el puente de comunicación esencial entre el frontend (desarrollado en React) y el backend (implementado en Django). Esta interfaz permite que los clientes interactúen con los datos y la lógica de negocio del sistema de manera estructurada, garantizando seguridad y escalabilidad.

En este proyecto, hemos desarrollado la API REST usando Django REST Framework, una herramienta robusta que facilita la creación de APIs seguras y eficientes en Python. Este enfoque permite que nuestro frontend interactúe fluidamente con el backend, realizando operaciones clave como consultar problemas, enviar soluciones, gestionar usuarios y automatizar la validación de código a través de peticiones HTTP estándar.

3.1.4.1. Serializers

Los *serializers* son componentes fundamentales en Django REST Framework que permiten transformar los datos complejos de los modelos de Django en formatos simples como JSON, adecuados para ser enviados a través de la API. Además, los serializers validan la información recibida desde el frontend antes de almacenarla en la base de datos, asegurando que los datos cumplen con los requisitos definidos. De este modo, los serializers actúan como un puente entre los modelos del backend y las peticiones/respuestas de la API, facilitando la comunicación y la integridad de los datos.

3.1.4.2. Views

Las *views* son las encargadas de gestionar las peticiones HTTP entrantes. Cada vista encapsula la lógica necesaria para procesar estas solicitudes: interactúan con los modelos y serializers, y luego devuelven una respuesta apropiada al cliente.

En este proyecto, hemos empleado tanto vistas basadas en clases como conjuntos de vistas (*ViewSet*s). Este enfoque nos permite una gestión eficiente y estructurada de los diversos recursos del sistema, como problemas, soluciones y usuarios. En esencia, las vistas determinan qué operaciones están permitidas en cada endpoint y cómo se aplican los permisos y la autenticación.

3.1.4.3. Endpoints principales

Un *endpoint* es una URL específica que expone la API REST, permitiendo a los clientes interactuar con un recurso concreto del sistema. Cada endpoint se asocia con uno o varios métodos HTTP (GET, POST, PUT, DELETE, etc.) para definir operaciones particulares, como obtener, crear, modificar o eliminar información.

- `/api/problemas/`: Gestión de problemas (listar, crear, obtener detalles, etc.).
- `/api/problemas/{id}/solutions/`: Envío y consulta de soluciones para un problema concreto.
- `/api/judge0-check/`: Validación automática de código mediante Judge0.

3.1.5. Autenticación y autorización

En este proyecto, hemos implementado un robusto sistema de gestión de usuarios y permisos, fundamental para garantizar una experiencia personalizada y segura. El sistema se basa en la autenticación mediante JSON Web Tokens (JWT). Además, se ha definido un esquema de roles diferenciados, que distingue entre profesorado y alumnado, asegurando así que cada tipo de usuario acceda únicamente a las funcionalidades que le corresponden. Esta arquitectura contribuye a mantener la integridad y la privacidad de los datos, al tiempo que permite una gestión eficiente y escalable de los permisos dentro de la plataforma.

3.1.5.1. Autenticación JWT

La autenticación JWT (JSON Web Token) permite a los usuarios iniciar sesión usando sus credenciales (nombre de usuario y contraseña). Si éstas son correctas, el servidor genera un token JWT. Este token contiene la información necesaria para identificar al usuario y su rol, y el backend lo verifica en cada solicitud para asegurar la autenticidad y los permisos.

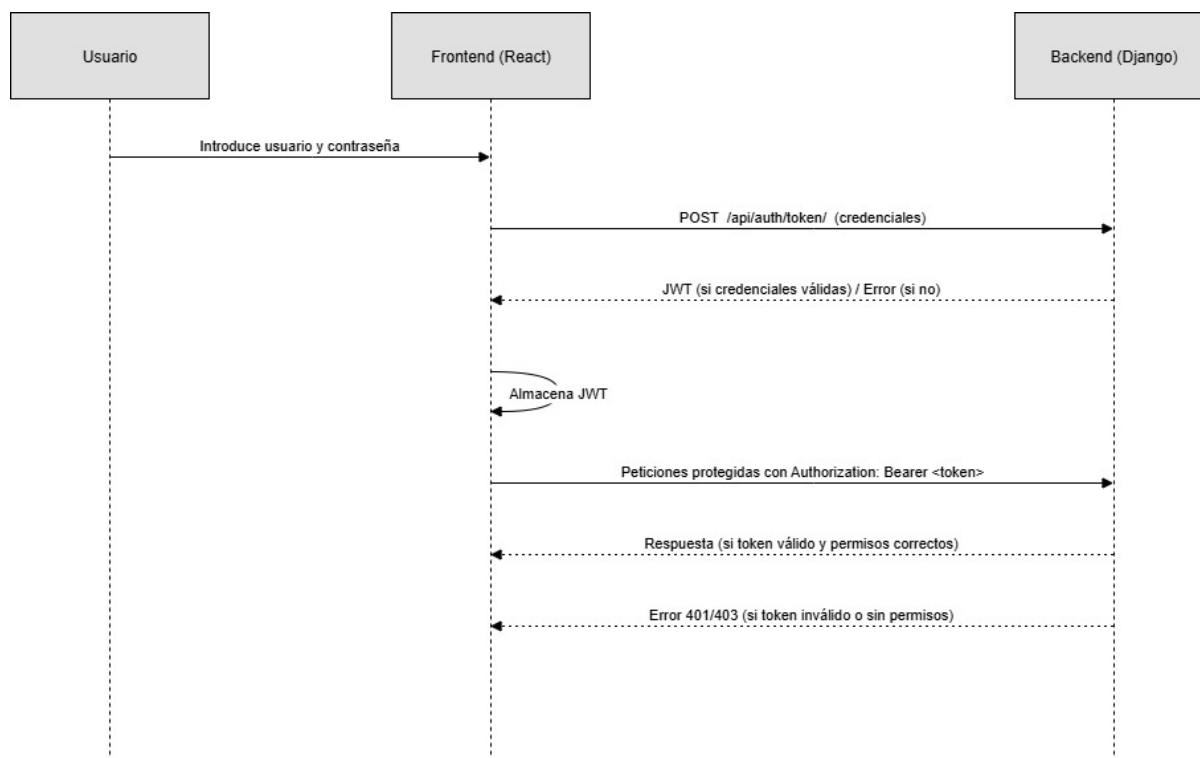


Figura 3.4: Flujo de autenticación con JWT

3.1.5.2. Autorización y control de acceso

El acceso a los distintos endpoints de la API está cuidadosamente restringido según el rol del usuario. Esto garantiza que cada tipo de usuario solo pueda realizar las acciones para las que tiene permiso:

- Solo los **profesores** pueden crear o modificar problemas.
- Únicamente los **alumnos** tienen permitido enviar soluciones a los problemas.
- Los **administradores** cuentan con acceso a funcionalidades avanzadas de gestión en toda la plataforma.

Además, existen endpoints públicos, como la consulta de problemas publicados, que están disponibles para cualquier usuario autenticado y no autenticado.

3.1.6. Judge0

Para la validación automática de las soluciones que los alumnos envían, el backend de nuestra aplicación se integra directamente con Judge0, un servicio externo. Judge0 es una API que nos permite ejecutar de forma segura código fuente en diversos lenguajes de programación y, además, comparar los resultados obtenidos con los resultados esperados para cada problema.

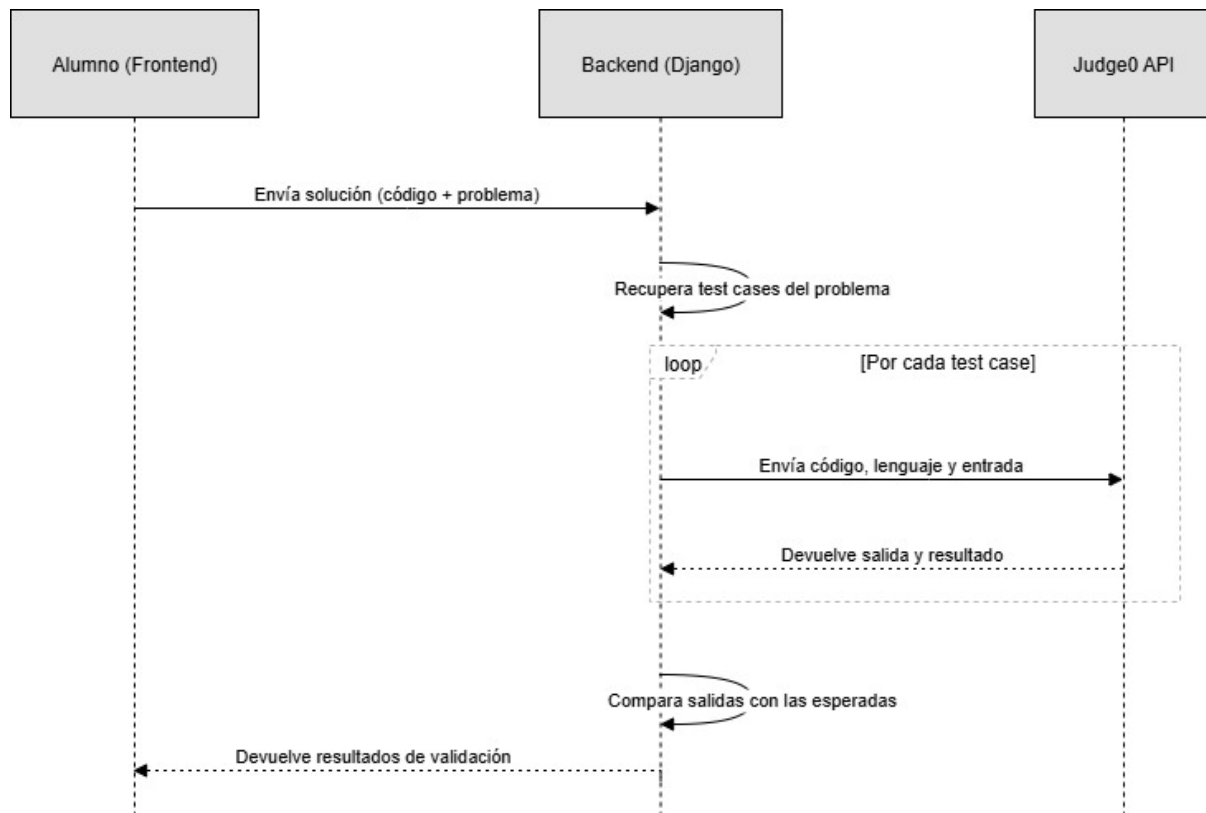


Figura 3.5: Integración del backend con Judge0

3.1.7. Configuración y Seguridad

El backend está configurado para garantizar la seguridad y el correcto funcionamiento de la API. Hemos habilitado la gestión de CORS para permitir una comunicación segura entre el frontend y el backend, incluso cuando están alojados en diferentes dominios.

Los endpoints sensibles están protegidos mediante autenticación JWT y un control de roles estricto. Esto garantiza que solo los usuarios autorizados puedan realizar operaciones críticas. Además, el sistema maneja los errores de forma robusta, devolviendo mensajes claros y códigos de estado HTTP apropiados, lo que facilita enormemente la detección y resolución de cualquier problema.

3.2. Frontend

En esta sección se explica todo lo relacionado con la interfaz de usuario, la experiencia de navegación y la comunicación con el backend. El frontend de este proyecto está desarrollado con React, una biblioteca de JavaScript que permite construir aplicaciones web dinámicas y modernas.

A través de esta interfaz intuitiva, los usuarios pueden:

- Autenticarse en la plataforma
- Consultar problemas
- Enviar soluciones
- Crear o editar problemas
- Visualizar los resultados de la validación automática de código

Todo esto se logra interactuando de forma sencilla y eficiente con la API REST del backend.

3.2.1. Tecnología y herramientas utilizadas

El frontend de este proyecto ha sido desarrollado utilizando React, una de las bibliotecas más populares para la construcción de interfaces de usuario dinámicas y eficientes en aplicaciones web. React permite construir componentes reutilizables, simplificando la gestión del estado y actualizando la interfaz de forma eficiente.

Para manejar las rutas y la navegación entre las diferentes páginas, usamos **React Router**. Esto nos permite ofrecer una experiencia de usuario fluida, sin necesidad de recargar la página por completo. La comunicación con el backend y con servicios externos se gestiona a través de la librería **Axios**, que simplifica las peticiones HTTP y el manejo de sus respuestas.

En cuanto a la gestión del estado global de la aplicación, se utiliza **Context API** de React. Esto nos permite compartir información relevante, como la autenticación del usuario, entre diferentes componentes de manera sencilla y eficiente.

Para el diseño y la presentación visual, se ha optado por el uso de **CSS personalizado**. Este está organizado en archivos específicos para cada componente o página, lo que facilita el mantenimiento y la escalabilidad del código.

3.2.2. Estructura del frontend

El frontend está organizado en una estructura de carpetas que facilita la modularidad y el mantenimiento del código. Los principales directorios y archivos son los siguientes:

- **src/components/**: Contiene los componentes reutilizables de la interfaz, como barras de herramientas, formularios y otros elementos visuales que se emplean en distintas partes de la aplicación.
- **src/pages/**: Incluye las páginas principales de la aplicación, cada una representando una vista o sección específica, como la página de inicio, el login o el detalle de un problema.

- **src/api/:** Aquí se centralizan las funciones y utilidades para la comunicación con el backend y servicios externos, como la autenticación y la interacción con Judge0.
- **src/context/:** Gestiona el contexto global de la aplicación, como la autenticación del usuario, permitiendo compartir información entre componentes de manera eficiente.
- **src/styles/:** Contiene los archivos CSS organizados por componente o página, lo que contribuye a mantener una estructura visual coherente en toda la aplicación y facilita tanto la personalización del diseño como el mantenimiento del código.

Esta organización permite separar claramente la lógica, la presentación y la gestión de datos, facilitando el desarrollo colaborativo y la escalabilidad del proyecto.

3.2.3. Comunicación con el backend

La comunicación entre el frontend y el backend se realiza principalmente a través de una API RESTful. Para gestionar las peticiones HTTP, se utiliza la librería Axios, que permite enviar solicitudes y recibir respuestas de manera sencilla y eficiente.

El frontend envía peticiones al backend para operaciones como el inicio de sesión, la obtención de problemas y soluciones, el envío de nuevas soluciones o la consulta de resultados de ejecución. Los datos intercambiados suelen estar en formato JSON, lo que facilita la integración y el procesamiento en ambos extremos.

En cuanto a la autenticación, el frontend gestiona los tokens JWT proporcionados por el backend tras el inicio de sesión. Estos tokens se almacenan de forma segura y se incluyen automáticamente en las cabeceras de las peticiones que requieren autorización, garantizando así la seguridad en el acceso a los recursos protegidos.

Además, el frontend también puede comunicarse con servicios externos, como Judge0, a través del backend, que actúa como intermediario para la ejecución y validación de código. De esta forma, se mantiene la seguridad y el control sobre las operaciones realizadas por los usuarios.

3.2.4. Principales componentes y páginas

El frontend está compuesto por una serie de componentes y páginas que estructuran la experiencia del usuario y permiten la interacción con las distintas funcionalidades de la aplicación. Entre los elementos más destacados se encuentran:

- **Home:** Página principal que da la bienvenida al usuario y puede mostrar información general o accesos directos a las funcionalidades más importantes.
- **LoginPage:** Página de inicio de sesión, donde los usuarios introducen su nombre de usuario y contraseña para acceder a la plataforma.
- **RegisterPage:** Página de registro que permite a los nuevos usuarios crear una cuenta proporcionando los datos requeridos, como nombre de usuario, correo electrónico, contraseña y rol (profesor o alumno), facilitando así su incorporación a la plataforma.

- **ProblemsPage:** Página que muestra el listado de problemas disponibles, permitiendo a los usuarios explorar, filtrar y seleccionar aquellos que desean resolver.
- **ProblemDetails:** Página de detalle de un problema específico, donde se muestra la descripción.
- **ProblemForm:** Página dedicada a la creación o edición de problemas, permitiendo a los usuarios con permisos adecuados añadir nuevos retos o modificar los existentes.
- **SubmitSolution:** Página destinada al envío de soluciones por parte de los usuarios, donde pueden cargar su código y recibir retroalimentación sobre su ejecución.
- **ComicToolbar:** Barra de herramientas que proporciona accesos rápidos y utilidades contextuales dentro de la interfaz, mejorando la experiencia de navegación y uso.

3.2.5. Gestión de estilos

La apariencia visual del frontend se gestiona principalmente mediante archivos **CSS** personalizados, organizados en la carpeta **src/styles/**. Cada página o componente relevante cuenta con su propio archivo de estilos, lo que permite mantener una separación clara entre la lógica y la presentación, así como facilitar el mantenimiento y la escalabilidad del diseño.

Esta organización permite aplicar estilos específicos a cada sección de la aplicación, asegurando una experiencia de usuario coherente y atractiva. El uso de clases y selectores bien definidos ayuda a evitar conflictos y a mantener el código CSS limpio y estructurado.

3.2.6. Apariencia y usabilidad

El diseño de la aplicación se ha concebido pensando en la **motivación y el atractivo** para un público adolescente, no necesariamente especializado en programación. Para ello, se ha optado por una estética inspirada en el **cómic**, utilizando colores vivos, tipografías desenfadadas y elementos visuales que recuerdan a las viñetas. Esta elección busca diferenciarse de las aplicaciones más formales y tradicionales, generando un entorno más **cercano, divertido y estimulante** para los usuarios jóvenes.

El lenguaje utilizado en la interfaz es directo y adaptado a su edad, evitando tecnicismos innecesarios y empleando expresiones cotidianas que facilitan la comprensión y la interacción. Por ejemplo, en lugar de mensajes como “Error de autenticación”, se muestra “¡Ups! Parece que tu usuario o contraseña no son correctos. ¿Lo intentamos de nuevo?”.

La **usabilidad** ha sido una prioridad en el diseño, asegurando que la navegación sea intuitiva y que las acciones principales estén siempre accesibles. Se han minimizado los pasos necesarios para realizar tareas clave, como registrarse, buscar problemas o enviar soluciones, y se han incorporado mensajes de ayuda y retroalimentación visual para guiar al usuario en todo momento.

En conjunto, estos elementos contribuyen a crear una experiencia atractiva, motivadora y accesible, pensada para enganchar a adolescentes de distintos perfiles y fomentar su participación activa en la plataforma.

3.2.7. Vista de la interfaz de usuario

A continuación se muestran varias capturas de pantalla de la aplicación, donde se puede apreciar el diseño final de la interfaz de usuario. Estas imágenes ilustran la apariencia, la organización de los elementos y la experiencia visual pensada para el público objetivo, destacando la estética basada en el cómic y la usabilidad adaptada a adolescentes.

¡BIENVENIDO A CODECHALLENGES!

Una plataforma para aprender programación resolviendo problemas reales de otras asignaturas. Profesores crean ejercicios. Estudiantes los resuelven.

VER RETOS

¡ÚNETE!

Figura 3.6: Página de bienvenida.

REGISTRO

Nombre de usuario *

Nombre *

Primer Apellido *

Segundo Apellido *

Email *

Contraseña *

Confirmar contraseña *

Rol *

REGISTRARSE

[¿Ya tienes una cuenta? Inicia sesión](#)

Figura 3.7: Registro

INICIAR SESIÓN

Nombre de usuario *
profe|

Contraseña *
.....

INICIAR SESIÓN

[¿No tienes una cuenta? Regístrate](#)

Figura 3.8: Inicio de sesión

[← VOLVER A PROBLEMAS](#)

CREAR NUEVO PROBLEMA

Título del problema *

Descripción del problema *

Ejemplo de entrada *

Ejemplo de salida *

Nivel educativo *

Dificultad *

Tiempo estimado (minutos) *
30

Referencia web (opcional)
URL opcional para más información

Materia
Matemáticas

[CREAR PROBLEMA](#) [CANCELAR](#)

Figura 3.9: Creación de un problema

PROBLEMAS DE PROGRAMACIÓN

Buscar problemas

Nivel

Dificultad

Materia

LIMPIAR FILTROS

ID	TÍTULO	NIVEL	DIFICULTAD	MATERIA	TIEMPO (MIN)	ACCIONES
7	Número más alto con dos dígitos	1º ESO	Fácil	Matemáticas	20	VER
8	Obtener la lista de cifras de un número	1º ESO	Fácil	Matemáticas	20	VER
9	Suma de las cifras de un número	1º ESO	Fácil	Matemáticas	30	VER
10	Comprobar si un dígito está en un número	1º ESO	Medio	Matemáticas	30	VER
11	Contar el número de cifras de un número	1º ESO	Fácil	Matemáticas	30	VER
12	Convertir una lista de dígitos en un número	1º ESO	Medio	Matemáticas	30	VER
13	Concatenar dos números	2º ESO	Fácil	Matemáticas	30	VER
14	Obtener la primera cifra de un número	1º ESO	Medio	Matemáticas	30	VER
15	Obtener la última cifra de un número	1º ESO	Medio	Matemáticas	30	VER
16	Número con cifras en orden inverso	1º ESO	Medio	Matemáticas	30	VER

Figura 3.10: Lista de problemas

[← VOLVER A PROBLEMAS](#)

FAHRENHEIT A CELSIUS

[✎ EDITAR](#)3º ESO Medio PUBLICADO 35 min Física

DESCRIPCIÓN

Vamos a empezar con algo sencillo para que te familiarices con cómo funciona el envío de código en Project Lovelace. Imagina que estás trabajando con un grupo de científicos que usan grados Celsius, pero las temperaturas que recibes están en Fahrenheit. Antes de poder hacer cualquier análisis, necesitas convertir esas temperaturas.

Tu tarea es escribir un programa que:

1. Lea una lista de temperaturas en Fahrenheit, una por línea, desde la entrada estándar.
2. Convierta cada temperatura a grados Celsius.
3. Muestre el resultado por pantalla como se muestra en el ejemplo de salida.

EJEMPLO DE ENTRADA

```
212
32
98.6
-40.0
```

EJEMPLO DE SALIDA

```
La temperatura en Celsius es: 100.0
La temperatura en Celsius es: 0.0
La temperatura en Celsius es: 37.0
La temperatura en Celsius es: -40.0
```

INFORMACIÓN ADICIONAL

ID del Problema	Creado por	Fecha de Creación	Última Modificación
18	Profesor	1 de julio de 2025, 19:53	1 de julio de 2025, 20:02

Figura 3.11: Vista de un problema

EJEMPLO DE ENTRADA

```
212
32
98.6
-40.0
```

EJEMPLO DE SALIDA

```
La temperatura en Celsius es: 100.0
La temperatura en Celsius es: 0.0
La temperatura en Celsius es: 37.0
La temperatura en Celsius es: -40.0
```

TU SOLUCIÓN

Lenguaje *

↑ SUBIR ARCHIVO DE CÓDIGO

CANCELAR

> ENVIAR SOLUCIÓN

Código

Figura 3.12: Vista del envío de una solución

Capítulo 4

Análisis de impacto

En este capítulo se analiza el impacto potencial de los resultados obtenidos durante la realización del TFG en diferentes contextos: personal, social y cultural. Se destacan tanto los beneficios esperados como los posibles efectos adversos, y se relaciona el trabajo con los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030.

4.1. Impacto personal

La realización de este Trabajo de Fin de Grado ha supuesto un importante crecimiento tanto a nivel académico como personal. Desde el punto de vista técnico, el desarrollo de la plataforma ha permitido profundizar en tecnologías actuales como React y Django, así como en conceptos clave de arquitectura de software, diseño de interfaces y gestión de bases de datos. El proceso ha implicado la adquisición de nuevas habilidades en programación *full-stack*, integración de servicios externos y despliegue de aplicaciones web, competencias altamente valoradas en el ámbito profesional.

La necesidad de documentar el trabajo, justificar decisiones técnicas y mantener una comunicación clara ha mejorado mis habilidades de redacción y presentación, fundamentales para el trabajo en equipo y la divulgación de resultados.

En el plano personal, el enfoque educativo y social del proyecto ha incrementado mi motivación, al percibir que el trabajo realizado puede tener un impacto positivo real en la formación de otros jóvenes. El reto de diseñar una interfaz atractiva y accesible para adolescentes ha estimulado mi creatividad y empatía, obligándome a ponerme en el lugar del usuario final y a buscar soluciones innovadoras y adaptadas a sus necesidades.

Además, la realización de este trabajo ha representado un valioso ejercicio de constancia y capacidad de adaptación. A lo largo del desarrollo, ha sido necesario superar diversos obstáculos técnicos, gestionar la incertidumbre y adaptarse a cambios tanto en los requisitos como en la visión inicial del proyecto. Enfrentar estos retos ha contribuido significativamente a reforzar mi confianza personal y profesional, así como a comprender la importancia de mantener una actitud flexible y proactiva ante los imprevistos.

4.2. Impacto Social

El impacto social de este proyecto es especialmente relevante, ya que su objetivo principal es democratizar el acceso a la educación en programación y fomentar el interés por la tecnología entre adolescentes, un grupo que a menudo se enfrenta a barreras de entrada en este ámbito. Al diseñar una plataforma accesible, atractiva y adaptada a su lenguaje y estética, se busca motivar a estudiantes de perfiles diversos, no solo a aquellos tradicionalmente interesados en la informática.

La colaboración prevista con institutos como el IES Las Musas (con el que la universidad ya colabora a través del tutor de este trabajo) permite que la aplicación se ajuste a las dinámicas reales del entorno escolar, facilitando la integración de la herramienta en el día a día de profesores y alumnado. Además, al ofrecer recursos y retos adaptados a distintos niveles y materias, se contribuye a reducir la brecha digital y a promover la igualdad de oportunidades en el acceso a competencias clave para el futuro profesional.

El proyecto también se inscribe en un proyecto de colaboración internacional. Se denomina "Socially transformative uses of logical AI as educational pivot", liderado por la profesora Veronica Dahl (Simon Fraser University, Canadá) y con la participación de PEG 2.0 consortium¹, CLAIRE² y la Universidad Politécnica de Madrid. El proyecto gira en torno al tema de la transición verde, atendiendo en paralelo a las transformaciones sociales necesarias para lograr simultáneamente la seguridad ecológica y social con, entre otras, actividades formativas para los jóvenes [25].

4.3. Objetivos de Desarrollo Sostenible

En septiembre de 2015, la Asamblea General de las Naciones Unidas adoptó la Agenda 2030 para el Desarrollo Sostenible, un plan de acción global a favor de las personas, el planeta y la prosperidad. Esta agenda establece 17 Objetivos de Desarrollo Sostenible (ODS) que abordan los principales desafíos sociales, económicos y ambientales a los que se enfrenta la humanidad, con el compromiso de no dejar a nadie atrás.

Los ODS incluyen metas como la erradicación de la pobreza, la mejora de la educación, la igualdad de género, la acción por el clima, la innovación tecnológica y la promoción de sociedades pacíficas e inclusivas. Cada objetivo está interrelacionado y requiere la colaboración de gobiernos, sector privado, sociedad civil y ciudadanía para su consecución.

En el contexto de este Trabajo de Fin de Grado, la plataforma desarrollada puede contribuir especialmente a los siguientes ODS:

- **ODS 4:** Educación de calidad. Facilita el acceso a recursos educativos innovadores y fomenta el aprendizaje de competencias digitales y de programación, esenciales en la sociedad actual.
- **ODS 5:** Igualdad de género. Promueve la participación igualitaria de estudiantes de todos los géneros en el ámbito STEM (Ciencia, Tecnología, Ingeniería y Matemáticas).

¹<https://prolog-lang.org/Education/>

²<https://cairne.eu/>

4.3. Objetivos de Desarrollo Sostenible

- **ODS 9:** Industria, innovación e infraestructura. Impulsa la innovación educativa mediante el uso de tecnologías digitales y la creación de infraestructuras de aprendizaje accesibles.
- **ODS 10:** Reducción de las desigualdades. Ofrece oportunidades de aprendizaje a estudiantes de diferentes contextos, contribuyendo a reducir la brecha digital.
- **ODS 17:** Alianzas para lograr los objetivos. Fomenta la colaboración entre instituciones educativas, profesorado y estudiantes para alcanzar metas comunes.

La integración de los ODS en proyectos tecnológicos y educativos es fundamental para avanzar hacia un desarrollo más justo, inclusivo y sostenible, alineando la innovación con los retos globales actuales. Todos ellos están presentes en el proyecto internacional mencionado.

Capítulo 5

Resultados y conclusiones

5.1. Resultados

Durante el desarrollo de este Trabajo de Fin de Grado, se ha implementado una aplicación web compuesta por un backend desarrollado en Django y un frontend en React. El sistema permite la gestión de usuarios, autenticación segura, y la interacción con problemas, integrando funcionalidades como la evaluación automática de soluciones mediante Judge0. Se han conseguido los siguientes resultados principales:

- **Backend robusto:** Se ha desarrollado una API RESTful utilizando Django, que permite la gestión eficiente de los recursos del sistema. El backend incorpora un sistema de roles y permisos personalizados, diferenciando entre administradores, profesores y estudiantes, y asegurando que cada usuario solo pueda acceder a las funcionalidades que le corresponden. Además, la arquitectura está pensada para ser segura y escalable, facilitando la incorporación de nuevas funcionalidades y el mantenimiento futuro del sistema.
- **Frontend funcional y atractivo:** El frontend, desarrollado con React, proporciona una interfaz de usuario moderna e intuitiva. Los usuarios pueden registrarse, iniciar sesión, consultar y resolver problemas, y visualizar sus resultados de manera clara y atractiva. Se ha puesto especial énfasis en la experiencia de usuario, asegurando una navegación fluida y accesible.
- **Integración de servicios externos:** Uno de los aspectos más relevantes ha sido la integración con la API de Judge0, que permite la evaluación automática de las soluciones enviadas por los usuarios. Esta funcionalidad automatiza la corrección de ejercicios de programación, proporcionando feedback inmediato.
- **Gestión de autenticación y sesiones:** Se ha implementado un sistema de autenticación basado en JSON Web Tokens (JWT), que representa una mejora significativa respecto a los métodos tradicionales de gestión de sesiones. El uso de JWT permite que, tras un inicio de sesión exitoso, el servidor genere un token firmado que contiene la información esencial del usuario y sus permisos. Este token es enviado al cliente y debe ser incluido en cada petición posterior, lo que elimina la necesidad de mantener sesiones en el servidor y reduce la carga sobre el backend.

5.2. Trabajo futuro

Aunque el sistema desarrollado cumple con los objetivos planteados, existen diversas líneas de trabajo futuro que podrían enriquecer y ampliar sus funcionalidades:

- **Ampliación de funcionalidades:** Se podría incorporar un sistema rankings, logros o estadísticas avanzadas que motiven la participación y el aprendizaje.
- **Internacionalización:** Traducir la aplicación a varios idiomas facilitaría su uso por parte de una comunidad más amplia y diversa.
- **Creación y gestión de grupos o clases:** Una mejora relevante sería permitir a los profesores crear y gestionar sus propios grupos de alumnos, a modo de clases virtuales. Esto facilitaría la organización de los estudiantes, la asignación de ejercicios personalizados a cada grupo y el seguimiento del progreso de los alumnos de manera más estructurada y eficiente.
- **Mejora del feedback sobre las soluciones enviadas:** Una posible línea de mejora sería ampliar el sistema de retroalimentación automática para que, además de comprobar la corrección funcional de las soluciones, también evalúe aspectos como la legibilidad del código, el uso de buenas prácticas de programación y la eficiencia de las soluciones. De este modo, los estudiantes recibirían comentarios no solo sobre si su código funciona, sino también sobre cómo pueden mejorar su estilo y evitar malos hábitos desde el principio. Esto ayudaría a corregir a tiempo posibles deficiencias en la escritura de código y fomentaría la adquisición de buenas prácticas desde las primeras etapas del aprendizaje.
- **Canal de comunicación para reporte de incidencias:** Implementar un sistema que permita a los profesores enviar mensajes o reportes directamente a los administradores de la plataforma en caso de detectar fallos, errores o incidencias en los problemas o en el funcionamiento general. Este canal serviría para mejorar la calidad y el mantenimiento de la plataforma, facilitando la detección temprana de errores y permitiendo a los administradores recibir feedback relevante de los usuarios con perfil docente. No sería necesario que estos mensajes reciban respuesta, ya que su objetivo principal es la notificación y el registro de incidencias para su posterior revisión y resolución.

Desde el punto de vista de la validación, la siguiente etapa consistirá en hacer una prueba piloto en un centro educativo (como ya se ha indicado, hay una colaboración arraigada con el IES Las Musas) que permita ajustar los elementos de la aplicación y, eventualmente, corregir posibles errores.

Tras este paso se pondría en funcionamiento publicando la página, tras anunciarla en redes y asociaciones de profesores de Tecnología (por ejemplo, APTE) y foros como AENUI.

Finalmente se quiere estudiar si la evaluación automática de soluciones mediante Judge0 es suficiente y funcional o si es preferible integrarla con Deliver-it, la aplicación de envío de prácticas de la ETSINF.

5.3. Conclusiones

La realización de este Trabajo de Fin de Grado ha supuesto una experiencia muy enriquecedora tanto a nivel técnico como personal. El desarrollo de una plataforma educativa de programación, compuesta por un backend robusto y seguro y un frontend intuitivo, me ha permitido aplicar y consolidar conocimientos adquiridos a lo largo de la carrera, así como aprender nuevas tecnologías y metodologías de trabajo.

A nivel técnico, el proyecto ha demostrado la viabilidad de integrar diferentes tecnologías (Django, React, Judge0, JWT) para construir un sistema completo, seguro y escalable. La implementación de roles y permisos, la integración de evaluación automática y la gestión eficiente de usuarios y ejercicios han sido retos superados con éxito, sentando una base sólida para futuras ampliaciones.

Desde el punto de vista personal, este trabajo me ha ayudado a mejorar mis habilidades de organización, planificación y resolución de problemas. He aprendido la importancia de la documentación y el control de versiones, así como a valorar la necesidad de diseñar sistemas pensando en la escalabilidad, la seguridad y la experiencia de usuario.


En definitiva, considero que el proyecto cumple con los objetivos planteados y aporta una solución útil y adaptable al ámbito educativo. Además, deja abiertas múltiples posibilidades de mejora y ampliación, lo que refuerza su valor como punto de partida para desarrollos futuros.

Bibliografía

- [1] Boletín Oficial del Estado. *Ley Orgánica 3/2020, de 29 de diciembre, por la que se modifica la Ley Orgánica 2/2006, de 3 de mayo, de Educación*. BOE-A-2020-17264. 2020. URL: <https://www.boe.es/eli/es/lo/2020/12/29/3>.
- [2] Comunidad de Madrid. *Decreto 65/2022, de 20 de julio, del Consejo de Gobierno de la Comunidad de Madrid*. wleg_pub - Comunidad de Madrid - madrid.org. 2022. URL: <https://www.comunidad.madrid/servicios/educacion/ordenacion-academica-eso>.
- [3] J. Waite y S. Sentance. *Teaching programming in schools: a review of approaches and strategies*. 2021. URL: <https://www.raspberrypi.org/app/uploads/2021/11/Teaching-programming-in-schools-pedagogy-review-Raspberry-Pi-Foundation.pdf>.
- [4] Michael Trucano. *Should All Students Learn How to Code? Pros and Cons*. WISE Qatar Foundation. 2022. URL: <https://www.wise-qatar.org/coding-cognitive-abilities-michael-trucano/>.
- [5] Mitchel Resnick et al. «Scratch: Programming for All». En: *Communications of the ACM* 52.11 (2009), págs. 60-67.
- [6] Niklas Humble. «A conceptual model of what programming affords secondary school courses in mathematics and technology». En: *Education and Information Technologies* 28.8 (ene. de 2023), págs. 10183-10208. ISSN: 1360-2357. DOI: 10.1007/s10639-023-11577-z. URL: <https://doi.org/10.1007/s10639-023-11577-z>.
- [7] Mara Saeli et al. «Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective». En: *Informatics in Education* 10 (abr. de 2011), págs. 73-88. DOI: 10.15388/infedu.2011.06.
- [8] Rimma Nyman, Kajsa Bråting y Cecilia Kilhamn and. «Can programming support mathematics learning? An analysis of Swedish lower secondary textbooks». En: *International Journal of Mathematical Education in Science and Technology* 56.7 (2025), págs. 1261-1279. DOI: 10.1080/0020739X.2024.2329345. eprint: <https://doi.org/10.1080/0020739X.2024.2329345>. URL: <https://doi.org/10.1080/0020739X.2024.2329345>.
- [9] Plataforma Europea de Educación Escolar. *Programación y programación informática en las escuelas: una tendencia emergente*. <https://school-education.ec.europa.eu/en/discover/news/computer-programming-and-coding-schools-emerging-trend>. Comisión Europea. 2023.
- [10] Educación 3.0. *Trabaja la programación en el aula con estos recursos*. <https://www.educaciontrespuntocero.com/noticias/programacion-en-el-aula/>. 2025.

- [11] S. Kim J. y Lee. «Bringing computational thinking to technology education classrooms: A case study from South Korea». En: *International Journal of Technology and Design Education* 33.1 (2022), págs. 1-19. DOI: 10.1007/s10798-022-09750-5. URL: <https://doi.org/10.1007/s10798-022-09750-5>.
- [12] Ministerio de Educación y Formación Profesional. *Competencias específicas de la asignatura Tecnología y Digitalización*. s.f. URL: <https://educagob.educacionfpydeportes.gob.es/curriculo/curriculo-lomloe/menu-curriculos-basicos/ed-secundaria-obligatoria/materias/tecnodigitali/competencias-especificas.html>.
- [13] Juan Ignacio Cabrera Hernández. «Nativos digitales que no lo son tanto». En: *Revista de Estudios de Juventud* 117 (2017), págs. 199-207.
- [14] Akdemia. *22 recursos para enseñar a tus alumnos a programar*. Consultado el 30 de junio de 2025. Ago. de 2023. URL: <https://www.akdemia.com/blog/22-recursos-para-ensenar-a-tus-alumnos-a-programar>.
- [15] EDUCACIÓN 3.0. *No sólo Scratch: 18 lenguajes y plataformas para enseñar programación en Primaria y Secundaria*. <https://www.educaciontrespuntocero.com/recursos/programacion/lenguajes-programacion-informatica-para-primaria-secundaria/>. Mayo de 2025.
- [16] *Project Lovelace: Problem-solving with science and code*. <https://projectlovelace.net/>.
- [17] *Codeforces: Competitive Programming Platform*. <https://codeforces.com/>.
- [18] Euler. *Project Euler*. <https://projecteuler.net/about>.
- [19] *¡Acepta el reto! Plataforma de programación competitiva en español*. <https://aceptaarelreto.com/>.
- [20] Jutge. *Jutge.org: The Virtual Learning Environment for Programming*. <https://jutge.org/>.
- [21] Django. *Django documentation*. <https://docs.djangoproject.com/en/stable/>.
- [22] React. *Home - Django REST framework*. <https://www.django-rest-framework.org/>.
- [23] JSON. *JSON Web Token Introduction - jwt.io*. <https://jwt.io/introduction>.
- [24] React. *Empezando - React*. <https://es.react.dev/learn/start-a-new-react-project>.
- [25] Veronica Dahl y Juan José Moreno-Navarro. «Doughnut Computing innbsp;City Planning fornnbsp;Achieving Human andnbsp;Planetary Rights». En: *Bio-Inspired Systems and Applications: From Robotics to Ambient Intelligence: 9th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2022*. Puerto de la Cruz, Tenerife, Spain: Springer-Verlag, 2022, págs. 562-572. ISBN: 978-3-031-06526-2. DOI: 10.1007/978-3-031-06527-9_56. URL: https://doi.org/10.1007/978-3-031-06527-9_56.

Este documento esta firmado por

	Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Wed Jul 02 18:17:53 CEST 2025
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)