

**Instituto Puig Castellar**

**Ciclo formativo:** Desarrollo de Aplicaciones Multiplataforma

**Grado:** 2º CFGS

**Crédito de Síntesis / Proyecto intermodular**

# INTO THE FOG

**Wiki interactiva sobre *Dead by Daylight***



**Autora:** Ainhoa Pedrosa Escudero

**Tutor:** Luis Elía Talón

**Curso:** 2025-2026

**Fecha de entrega:** 15/05/2026

## **Resumen del proyecto**

El proyecto Into The Fog trata de una página web basada en el videojuego Dead by Daylight. El objetivo es combinar una wiki informativa en forma de web completa y funcional, con un sistema de minijuegos. Los jugadores tendrán un espacio en el que pueden consultar información detallada sobre el juego y poner a prueba su nivel de conocimiento.

A nivel técnico, quiero aplicar los conocimientos adquiridos en el curso, como desarrollo del backend con Spring Boot, un frontend con Vue y gestión de base de datos con PostgreSQL.

Para llevar a cabo el proyecto he dividido el proyecto en diferentes fases:

- Definir las funcionalidades principales de la web y lo que debe hacer
- Crear la base de datos con la información del juego y usuarios
- Desarrollar el backend con Spring Boot
- Crear el frontend con Vue
- Con la información en la web, crear el minijuego
- Hacer pruebas para comprobar el funcionamiento

El proyecto tiene que demostrar la capacidad de crear una web completa integrando el backend, frontend y base de datos. El desarrollo del proyecto tiene que mejorar esos conocimientos técnicos, la planificación y obtener experiencia.

## **Palabras clave**

- Dead by Daylight
- Aplicación web
- Wiki de videojuegos
- Minijuego diario
- Spring Boot
- Vue
- PostgreSQL

## **Abstract**

The Into The Fog project is a website based on the video game Dead by Daylight. The goal is to combine an informative wiki in the form of a complete and functional website with a system of mini-games. Players will have a space where they can consult detailed information about the game and test their knowledge.

On a technical level, I want to apply the knowledge I have acquired in the course, such as backend development with Spring Boot, frontend with Vue, and database management with PostgreSQL.

To carry out the project, I have divided it into different phases:

- Define the main functionalities of the website and what it should do.
- Create the database with game and user information.
- Develop the backend with Spring Boot.
- Create the frontend with Vue.
- With the information on the website, create the mini-games
- Run tests to check that everything is working

The project must demonstrate the ability to create a complete website integrating the backend, frontend, and database. The development of the project must improve technical knowledge, planning, and gain experience.

## **Keywords**

- Dead by Daylight
- Web application
- Gaming wiki
- Daily mini game
- Spring Boot
- Vue
- PostgreSQL

# Índice

<b>1. Presentación del proyecto.....</b>	<b>5</b>
1.1 Introducción.....	5
1.2 Contexto.....	5
1.3 Justificación.....	6
1.4 Objetivos.....	7
<b>2. Estrategia y planificación.....</b>	<b>8</b>
2.1 Estrategia de desarrollo y viabilidad.....	8
2.2 Metodología de trabajo.....	8
2.3 Planificación.....	9
<b>3. Análisis.....</b>	<b>11</b>
3.1 Casos de uso.....	11
3.2 Requisitos funcionales.....	12
3.3 Requisitos no funcionales.....	12
3.4 Análisis de alternativas tecnológicas.....	13
<b>4. Diseño.....</b>	<b>14</b>
4.1 Arquitectura del sistema.....	14
4.2 Modelo de datos.....	14
4.3 Diseño de interfaz.....	15
<b>5. Desarrollo.....</b>	<b>17</b>
5.1 Estructura del proyecto.....	17
5.2 Implementación de funcionalidades.....	17
5.3 Pruebas.....	20
<b>6. Conclusiones.....</b>	<b>22</b>
6.1 Conclusiones generales.....	22
6.2 Consecución de objetivos.....	22
6.3 Valoración de la metodología y planificación.....	23
6.4 Visión de futuro.....	23
<b>7. Glosario.....</b>	<b>25</b>
<b>8. Bibliografía.....</b>	<b>27</b>
<b>9. Anexos.....</b>	<b>28</b>

# 1. Presentación del proyecto

## 1.1 Introducción

Este proyecto trata del desarrollo de una página web ambientada en el videojuego Dead by Daylight, donde la idea principal es crear una web que funcione como un tipo de wiki para que los usuarios puedan consultar información sobre los asesinos, supervivientes, mapas y objetos del juego, pero añadiendo también un apartado con minijuegos y novedades para hacer la experiencia más entretenida.

Hoy en día la información sobre el juego está repartida en diferentes páginas como wikis o foros que solo tienen la opción de mostrar la información. Con este proyecto pretendo mejorar eso creando una web más ordenada, con la información necesaria para el videojuego y que además de consultar esa información puedas interactuar y jugar, y así darle un toque interactivo. Además, la web incorpora un foro donde los usuarios registrados pueden crear temas y comentar para compartir opiniones y experiencias sobre el juego.

La web está dividida en dos partes: por un lado el backend desarrollado con Spring Boot, que se encarga de gestionar los datos y “servirlos”. Por el otro lado está el frontend, desarrollado con Vue para mostrar la información y permitir la interacción.

El objetivo principal es crear una web clara y fácil de usar que combine información esencial del juego con minijuegos.

## 1.2 Contexto

El proyecto se centra dentro del ámbito del desarrollo de aplicaciones web relacionadas con videojuegos, en concreto con Dead by Daylight, un videojuego multijugador que ha sido bastante popular y con una gran comunidad de jugadores.

La información sobre el juego está sobre todo en páginas tipo wiki de fandoms o en la oficial, siendo muy útiles para consultar datos pero que tienden a ser bastante densas y

poco intuitivas ya que hay mucha información en ellas que no es tan relevante a la hora de jugar las partidas.

En los últimos años han surgido páginas web centradas en una temática concreta, por ejemplo, en un videojuego donde van a poner a prueba tus conocimientos sobre este con diferentes minijuegos. Además, la web incorpora una parte más social mediante un foro donde los usuarios pueden interactuar, crear temas y compartir opiniones relacionadas con el videojuego.

Mi idea para este proyecto sale de estos dos tipos de páginas webs, ya que como jugadora del videojuego las visitaba a menudo y hasta día de hoy no he encontrado una web que lo integrase todo en un mismo lado.

En este contexto surge la idea de hacer una página web que unifique estos dos aspectos: una wiki con información clara y organizada del videojuego, junto con minijuegos que permitan al usuario poner a prueba sus conocimientos. Se mejoraría la experiencia del usuario ya que se ofrece una alternativa más completa.

### **1.3 Justificación**

El sentido de este proyecto está principalmente en que hoy en día no hay ninguna página que combine de forma clara y sencilla tanto la información del videojuego como una parte más interactiva con pruebas. Hay wikis con muchísima información pero yo siempre les encuentro el mismo problema y es que no son muy claras, y por supuesto que los minijuegos están en páginas separadas haciendo que la experiencia del usuario que quiera usar ambos tipos sea menos cómoda.

El problema principal que quiero resolver es esa separación entre información y entretenimiento. La mejora que aporta este proyecto es la unificación de ambos conceptos en una sola aplicación, así de esta manera el usuario puede tener a su alcance toda la información relevante y participar en minijuegos relacionados sin tener que salir de la web. También se añade una parte más social mediante un foro donde los usuarios registrados pueden compartir opiniones, crear temas y comentar sobre el videojuego.

En cuanto a la utilidad, el proyecto puede servirle tanto a jugadores nuevos, como a aquellos más avanzados que quieran poner a prueba los conocimientos que tienen de Dead by Daylight.

El proyecto tiene un valor importante a nivel de aprendizaje, ya que me permite aplicar conocimientos de desarrollo web, como la creación de una API con Spring Boot y el desarrollo de una interfaz interactiva con Vue. También tendré que trabajar con bases de datos, cómo organizar la información y establecer la conexión entre el backend y el frontend.

## 1.4 Objetivos

El objetivo general de mi proyecto es desarrollar una página web interactiva basada en Dead by Daylight que permita a los usuarios consultar información sobre asesinos, supervivientes, objetos, mapas y que tengan una parte interactiva gracias al minijuego diario.

Como objetivos más específicos tengo los siguientes:

- Implementar una web intuitiva y moderna.
- Desarrollar un sistema de navegación claro que permita acceder fácilmente a distintas secciones.
- Crear un backend que gestione los datos del juego mediante una API REST que permite acceder a la información de la base de datos.
- Desarrollar minijuegos en el que el usuario debe adivinar que asesino es con pistas visuales.
- Implementar un sistema de usuarios y un foro donde los jugadores puedan crear temas y comentar.
- Implementar persistencia en el cliente mediante almacenamiento local para mantener el progreso del usuario en los minijuegos.
- Permitir la personalización del perfil de usuario y el guardado de personajes favoritos.

## **2. Estrategia y planificación**

### **2.1 Estrategia de desarrollo y viabilidad**

El desarrollo del proyecto se plantea mediante una arquitectura cliente-servidor, separando el frontend y el backend para facilitar la comprensión del código.

El frontend se desarrollará utilizando Vue, creando una interfaz dinámica y clara. El backend se implementará con Spring Boot para gestionar toda la lógica y comunicación con la base de datos mediante una API REST.

Además, el backend también se encargará de gestionar funcionalidades como el sistema de foro, perfiles de usuario y el progreso guardado del minijuego.

Desde el punto de vista técnico, el proyecto es viable ya que usa tecnologías que hemos utilizado en el curso y que están bien documentadas.

A nivel temporal, el desarrollo lo he organizado en fases progresivas (diseño, implementación y pruebas) para así avanzar de forma clara y controlada, detectando errores de forma temprana.

En cuanto a los recursos, solo necesito un entorno de desarrollo (en este caso IDE, navegador, servidor local y gestor de base de datos) lo que hace que el proyecto sea viable y sin ningún coste o dificultad.

### **2.2 Metodología de trabajo**

La metodología que he utilizado para el desarrollo del proyecto es de tipo iterativo e incremental. El proyecto se ha dividido en pequeñas funcionalidades que he ido implementando progresivamente para así ir validando cada una de ellas antes de avanzar.

Ha facilitado la detección de errores, la mejora continua del diseño y la adaptación a posibles cambios.

El orden que sigo es:

- Análisis de la funcionalidad que quiero implementar
- Desarrollo del código
- Pruebas del funcionamiento
- Ajustes o mejoras

## 2.3 Planificación

El desarrollo del proyecto lo he organizado en diferentes apartados:

### Fase 1: Análisis y diseño

- Definición de los objetivos del proyecto
- Diseño de la base de datos
- Definición de la arquitectura cliente-servidor
- Planificación de las vistas principales

### Fase 2: Desarrollo del backend

- Creación de entidades
- Implementación de repositorios
- Desarrollo de controladores REST
- Exposición de endpoints
- Pruebas de funcionamiento de la API

### Fase 3: Desarrollo del frontend

- Creación de las vistas principales
- Integración con el backend
- Diseño de la interfaz
- Implementación de navegación (router)

### Fase 4: Implementación del minijuego

- Desarrollo del sistema para mostrar los asesinos

- Implementación de sugerencias
- Sistema de autocompletado en el buscador
- Persistencia del estado
- Sistema de progreso guardado para cada usuario
- Navegación entre niveles y visualización de niveles completados

#### **Fase 5: Funcionalidades adicionales**

- Sistema de autenticación de usuarios
- Perfil de usuario personalizable
- Sistema de favoritos
- Implementación del foro con temas y comentarios
- Restricciones de acceso para usuarios no autenticados

#### **Fase 6: Pruebas y mejoras**

- Corrección de errores
- Optimización del rendimiento
- Ajustes visuales

## 3. Análisis

### 3.1 Casos de uso

- **Consultar asesinos:** Que el usuario pueda acceder al listado de asesinos y ver la información general de cada uno.
- **Consultar información detallada sobre asesinos:** Poder seleccionar un asesino y ver toda la información de este.
- **Consultar supervivientes:** Que el usuario pueda acceder al listado de supervivientes y ver la información general de cada uno.
- **Consultar información detallada sobre supervivientes:** Poder seleccionar un superviviente y ver toda la información de este.
- **Consultar mapas:** El usuario debe poder ver los mapas que hay disponibles en el videojuego y sus variantes.
- **Consultar objeto:** Que pueda ver los objetos que hay disponibles y su descripción.
- **Navegar entre secciones:** El usuario podrá desplazarse entre las diferentes vistas por la barra de navegación.
- **Registrarse:** Que se pueda registrar el usuario proporcionando nombre de usuario, mail y contraseña.
- **Iniciar sesión:** Que pueda autenticarse en la web.
- **Gestionar perfil:** Una vez autenticado, que pueda acceder a su perfil y consultar su información.
- **Guardar favoritos:** Que pueda marcar como favoritos distintos elementos de la web (supervivientes, asesinos, objetos o mapas).
- **Jugar al minijuego:** Que el usuario pueda acceder a los diferentes niveles del minijuego e intentar adivinar el asesino correcto mediante pistas visuales.
- **Guardar progreso del minijuego:** Que el usuario autenticado pueda conservar los niveles completados y visualizar su progreso.
- **Participar en el foro:** Que el usuario pueda crear temas y comentar publicaciones dentro del foro.

## 3.2 Requisitos funcionales

- Debe mostrar una página principal desde la que se pueda acceder a diferentes secciones de la web.
- Debe permitir consultar listados de asesinos, supervivientes, objetos y mapas.
- Debe mostrar información detallada de cada asesino y superviviente.
- Debe permitir visualizar mapas junto con sus variantes.
- Debe mostrar objetos con su descripción.
- Debe obtener toda la información desde una base de datos.
- Debe disponer de una barra de navegación funcional.
- Debe permitir el registro de nuevos usuarios.
- Debe validar los datos introducidos en el registro.
- Debe prohibir el registro a usuarios con correos o nombres duplicados.
- Debe permitir el inicio de sesión de usuarios registrados.
- Debe dejar añadir elementos a favoritos al usuario registrado.
- Debe permitir la creación y edición de perfiles de usuario.
- Debe incluir un foro donde los usuarios registrados puedan crear temas y comentar publicaciones.
- Debe impedir crear temas o comentarios si el usuario no ha iniciado sesión.
- Debe incluir un minijuego interactivo basado en emojis relacionados con los asesinos.
- Debe guardar el progreso del usuario en el minijuego.
- Debe mostrar visualmente los niveles completados del minijuego.
- Debe permitir navegar entre los diferentes niveles del minijuego.

## 3.3 Requisitos no funcionales

- La web debe contar con una interfaz clara, intuitiva y fácil de utilizar, para que cualquier usuario pueda navegar entre secciones sin dificultad.
- La interfaz debe adaptarse a diferentes tamaños de pantalla para permitir el uso en ordenadores, móviles o tablets.

- La carga de los datos debe realizarse de forma eficiente para evitar tiempos de espera muy largos.
- El sistema debe mantener correctamente el progreso y estado del minijuego para cada usuario.
- El código tiene que estar bien estructurado para que sea fácil mejorarlo o hacer correcciones.
- El sistema debe proteger adecuadamente la información de los usuarios y evitar accesos indebidos.
- Que la aplicación esté disponible siempre que el servidor esté activo.

### **3.4 Análisis de alternativas tecnológicas**

Para el frontend consideré HTML, CSS y JavaScript puro ya que eran las opciones más básicas pero se quedaban un poco cortas para una web con diferentes páginas y con componentes reutilizables. También consideré React pero era más complicada de usar y no la habíamos utilizado en las clases. Por eso finalmente me decanté por Vue, que es más fácil de aprender, permite organizar mejor el código y la ibamos usando en algunas asignaturas.

Para el backend elegí directamente Spring Boot ya que era más fácil de usar con bases de datos y por último, para la base de datos elegí PostgreSQL.

La comunicación entre el frontend y backend se hace a través de una API REST. Las herramientas de desarrollo que he usado han sido VSCode, PgAdmin y IntelliJ IDEA.

## 4. Diseño

### 4.1 Arquitectura del sistema

El sistema sigue una arquitectura cliente-servidor, donde se separa la parte visual de la lógica.

- **Frontend (Vue):** Se encarga de la interfaz de usuario, donde este puede navegar por la página, registrarse, iniciar sesión y ver información. También gestiona funcionalidades interactivas como el foro, el sistema de favoritos y el minijuego.
- **Backend (Spring Boot):** Gestiona la lógica del sistema, como el registro e inicio de sesión de usuarios. También proporciona una API REST para que el frontend pueda tener acceso a los datos y enviarlos. Además, controla el sistema de progreso del minijuego, los comentarios del foro y la gestión de perfiles de usuario.
- **Base de datos (PostgreSQL):** Se almacena toda la información relacionada con usuarios, personajes, objetos, mapas, favoritos, temas del foro, comentarios y progreso del minijuego.

### 4.2 Modelo de datos

El sistema utiliza una base de datos relacional con varias tablas principales.

- killer\_emojis: killer\_id, emoji\_1, emoji\_2, emoji\_3
- killers: id, apodo, nombre, imagen\_killer, lore, radio\_terror, poder, imagen\_poder, descripcion\_poder, fecha, genero, nacionalidad, arma, velocidad, altura, nombre\_habilidad\_1, nombre\_habilidad\_2, nombre\_habilidad\_3, habilidad\_1, habilidad\_2, habilidad\_3, imagen\_habilidad\_1, imagen\_habilidad\_2, imagen\_habilidad\_3, dificultad, tipo\_ataque
- mapas: id, nombre
- mapas\_variantes: id, mapa\_id, nombre, imagen
- objetos: id, nombre, icono, descripcion

- survis: id, nombre, imagen\_survi, lore, fecha, genero, nacionalidad, nombre\_habilidad\_1, nombre\_habilidad\_2, nombre\_habilidad\_3, habilidad\_1, habilidad\_2, habilidad\_3, imagen\_habilidad\_1, imagen\_habilidad\_2, imagen\_habilidad\_3
- usuarios: id, nombre\_usuario, email, password
- perfil\_usuario: id, usuario\_id, avatar, nombre\_juego, year\_inicio, descripcion
- favoritos: id, usuario\_id, personaje\_id, tipo
- temas: id, titulo, contenido, usuario\_id, fecha
- comentarios: id, contenido, usuario\_id, tema\_id, fecha
- killer\_game\_progress: id, usuario\_id, killer\_id, completado

Relaciones:

- Un usuario puede tener un perfil 1:1
- Un usuario puede tener varios favoritos 1:N
- Un tema puede tener varios comentarios 1:N
- Un usuario puede crear varios temas y comentarios 1:N
- Un usuario puede completar varios niveles del minijuego 1:N
- Un mapa puede tener varias variantes 1:N
- Un killer tiene asociado un conjunto de emojis para el minijuego 1:1

### 4.3 Diseño de interfaz

La interfaz está diseñada con un estilo oscuro y uniforme en relación al videojuego, manteniendo la coherencia en toda la web.

Las características principales a destacar son por ejemplo, el uso de colores oscuros, un diseño centrado y simple, los elementos con bordes redondeados, con una navegación mediante un navbar superior y siendo la interfaz responsive. También se han diseñado interfaces específicas para el foro y el minijuego, manteniendo la misma estética oscura y uniforme del resto de la web.

El usuario puede acceder a la información de los asesinos, supervivientes, mapas y objetos, además del sistema de registro, inicio de sesión, perfiles de usuario, foro y minijuego

interactivo. La web está diseñada con formularios simples, validaciones para evitar errores, mensajes de error/éxito y con un cambio dinámico del navbar (login).



*La imagen muestra la interfaz principal de la aplicación con el diseño temático inspirado en Dead by Daylight. El resto de capturas completas pueden consultarse en el Anexo.*

# 5. Desarrollo

## 5.1 Estructura del proyecto

El proyecto está dividido en el frontend y el backend, siguiendo una estructura cliente-servidor. El frontend está desarrollado con Vue.js, que se encarga de la interfaz visual y la interacción con el usuario.

La estructura principal del frontend es:

- **Views:** Aquí están las páginas principales de la web, como la pantalla de inicio, el perfil de usuario, listados de personajes... También incluye vistas para el foro, el minijuego y los niveles del juego.
- **Components:** Son los componentes reutilizables, como por ejemplo el navbar.
- **Router:** Para configurar las rutas de la web.
- **Public/ img:** Aquí guardo las imágenes utilizadas, como en el avatar, los personajes, items y emojis utilizados en el minijuego.

El backend está desarrollado con Spring Boot y se encarga de la lógica y el acceso a los datos.

La estructura principal sería la siguiente:

- **Repository:** Son interfaces que extienden JpaRepository para que pueda acceder a la base de datos de forma automática.
- **Controller:** Guardo los controladores que se encargan de recibir las peticiones del frontend y devolver respuestas.
- **Entity:** Aquí he definido las entidades que representan las tablas de la base de datos. También gestiona funcionalidades como el foro, perfiles de usuario, favoritos y progreso del minijuego.

La base de datos está creada con PostgreSQL.

## 5.2 Implementación de funcionalidades

- **Perfil de usuario:** Permite a cada usuario crear y editar su perfil.

Al acceder al perfil, el frontend realiza una petición al backend para comprobar si el usuario ya tiene un perfil creado, y si existe muestra los datos guardados. Si no existe se muestra un formulario para completarlo. Los datos que almacena son: nombre en el juego, avatar seleccionado, año de inicio, personajes favoritos y una descripción personal.

He definido dos endpoints principales que obtiene los datos y el otro los guarda. El endpoint de tipo GET recupera la información del perfil del usuario a partir de un id. El endpoint POST se encarga de guardar la información del perfil, recibe los datos desde el frontend y decide si crear un nuevo perfil o actualizarlo si ya existe.

- **Sistema de favoritos:** Permite a cada usuario marcar personajes como favoritos.

En la vista individual de cada personaje aparece un botón con forma de corazón, que indica si ese personaje está ya marcado o no y aparece en el perfil.

El frontend envía una petición al backend para comprobar si el personaje ya está marcado como favorito y cuando el usuario interactúa se actualiza ese valor. He definido endpoints que permiten obtener los favoritos de cada usuario según el tipo (asesino o superviviente) y otro que gestiona la acción de añadir o eliminar a partir de toggle, es decir, si el favorito ya existe se elimina y si no existe se añade.

- **Favoritos en el perfil:** Para que aparezcan los personajes favoritos en el perfil.

Los personajes favoritos no se introducen manualmente en el perfil, se generan automáticamente usando los datos de la base de datos. Al cargar el perfil el frontend realiza peticiones al backend para obtener los favoritos del usuario, consulta la información de ese personaje para mostrar el nombre e imagen en el perfil.

- **Visualización de personajes, objetos y mapas:** Permite visualizar un listado de estos datos.

El frontend realiza peticiones al backend para obtener todos los datos de personajes, objetos y mapas y así mostrarlos. Al seleccionar un personaje, se accede a una vista individual con la información relevante almacenada en la base de datos.

He definido endpoints de tipo GET para obtener el listado completo de cada personaje o dato.

- **Sistema de novedades:** Un sistema de noticias en la página principal.

Las novedades están almacenadas en la base de datos y el frontend realiza una petición al backend para obtenerlas y ordenadas por fecha, para mostrar siempre las más recientes.

He definido un endpoint de tipo GET para que devuelva todas las novedades. Se muestran en un carrusel en la pantalla de inicio permitiendo navegar entre ellas a partir de unas flechas. El frontend gestiona qué noticia se muestra utilizando una variable que controla la posición actual dentro de la lista de novedades, cada vez que el usuario pulsa una flecha se actualiza la posición.

- **Foro:** Permite a los usuarios registrados crear temas y comentar publicaciones.

El frontend obtiene la lista de temas desde el backend y muestra información como el título, contenido, usuario y fecha de publicación. Al acceder a un tema concreto se cargan automáticamente todos los comentarios asociados.

He definido endpoints GET para obtener los temas y comentarios, y endpoints POST para crear nuevos temas y comentarios. Además, el sistema comprueba si el usuario ha iniciado sesión antes de permitir publicar o comentar.

- **Minijuego:** Se ha desarrollado un minijuego interactivo donde el usuario debe adivinar el asesino correcto utilizando tres emojis como pista visual.

El sistema muestra diferentes niveles numerados y permite navegar entre ellos. El frontend obtiene desde el backend los emojis correspondientes a cada asesino y muestra un buscador con autocompletado para seleccionar la respuesta.

Cuando el usuario selecciona un asesino incorrecto, este queda registrado visualmente en rojo. Si acierta, el asesino aparece marcado en verde y el progreso queda guardado. También se almacena el estado de cada nivel mediante almacenamiento local y una tabla de progreso en la base de datos para mantener los niveles completados por cada usuario.

- **Interfaz de usuario:** He desarrollado una interfaz con un diseño sencillo.

He utilizado un tema oscuro y una estructura organizada a partir de componentes reutilizables, como por ejemplo el navbar. He aplicado estilos con CSS para mejorar la visualización, incluyendo efectos como fondos semitransparentes, hover en botones... La interfaz la he pensado para que sea clara, intuitiva y adaptable a diferentes tamaños de pantalla.

## 5.3 Pruebas

- **Gestión de perfiles:** He comprobado que el perfil se crea correctamente cuando el usuario introduce los datos por primera vez. También que al acceder de nuevo al perfil la información se ha guardado bien y se muestra.
- **Sistema de favoritos:** He probado el botón en las vistas de los personajes y que al presionarlo cambie el estado. También se ha comprobado que aparezca en el perfil si está marcado y un máximo de 3 por tipo.
- **Visualización de personajes:** He revisado que el listado de personajes se carga correctamente desde el backend y que muestra la información correcta.
- **Novedades:** He probado que se muestren todas las noticias de la base de datos y que la información esté entera y correcta. También que el carrusel deje navegar entre las noticias que hay disponibles.
- **Foro:** He comprobado que los usuarios registrados pueden crear temas y comentarios correctamente. También he probado que los usuarios sin iniciar sesión no puedan publicar contenido y únicamente puedan visualizar el foro.
- **Minijuego:** He probado que los niveles carguen correctamente los emojis asociados a cada asesino y que el sistema de búsqueda muestre sugerencias válidas. También he comprobado que los intentos fallidos se muestren correctamente, que el asesino

acertado aparezca marcado en verde y que el progreso quede guardado incluso al volver a entrar en el nivel.

- **Pruebas en la interfaz:** He hecho pruebas para comprobar que se vea bien en diferentes resoluciones y que funcionen correctamente los elementos interactivos, los hovers, botones, formularios y la navegación entre páginas.

He probado también situaciones como que los datos no existan o que haya fallos de conexión con el backend y que en estos casos la web gestione correctamente los errores a partir de los try/catch para evitar fallos mayores.

## **6. Conclusiones**

### **6.1 Conclusiones generales**

Llevar a cabo este proyecto me ha permitido saber crear una web completa, centrándose en unos de mis videojuegos favoritos, ampliando también a su vez mi conocimiento de este. A lo largo de todo el proceso he implementado muchas funcionalidades, como la visualización de personajes, gestión de perfiles, sistema de favoritos, sección de novedades y un foro interactivo. También he desarrollado un minijuego con sistema de progreso y almacenamiento del estado de cada nivel.

Uno de los principales logros ha sido poder conectar y que se comunique bien el frontend con el backend mediante peticiones HTTP, además de conseguir implementar correctamente el minijuego interactivo y mantener una interfaz coherente con el estilo general de la web. También me ha permitido aplicar los conocimientos del manejo de base de datos.

En conclusión, he logrado desarrollar una web funcional, estable y completa que cumple con los requisitos y que también tiene margen para mejoras o actualizaciones.

### **6.2 Consecución de objetivos**

Los objetivos que me planteé al iniciar el proyecto los he cumplido, ya que he conseguido desarrollar una web funcional basada en una arquitectura cliente-servidor.

En primer lugar, he conseguido implementar un sistema completo, compuesto por un frontend desarrollado con Vue y un backend basado en Spring Boot, permitiendo la comunicación entre los dos mediante una API REST.

He desarrollado también varias funcionalidades clave: la visualización de los diferentes personajes, objetos y mapas, un sistema de perfiles de usuario personalizables y sistema de favoritos. También he desarrollado un foro en el que los usuarios pueden crear temas y comentar en ellos, aplicando una restricción en la que los usuarios no registrados no pueden

comentar. Además, he implementado un minijuego interactivo basado en emojis, con navegación entre niveles, sistema de progreso y almacenamiento del estado de cada partida.

He conseguido mantener una interfaz clara y coherente en toda la web, que esté acorde con la temática y que sea intuitiva y fácil de usar.

En conclusión, los objetivos que me propuse al inicio los he cumplido y he ido añadiendo mejoras para mejorar el resultado final del proyecto.

### **6.3 Valoración de la metodología y planificación**

El desarrollo del proyecto lo he llevado a cabo de forma progresiva, trabajando paso a paso y añadiendo las funcionalidades despacio en lugar de hacerlo todo de golpe, primero he creado las partes más básicas y después he ido mejorando y ampliando.

Esta forma de trabajo ha sido útil porque he podido comprobar que cada parte funcionaba correctamente antes de seguir avanzando. Cuando aparecía algún error era más fácil detectarlo y solucionarlo. También ha ido bien a la hora de conectar el backend con el frontend ya que he ido probando cada petición que añadía constantemente.

La planificación inicial ha servido como guía, aunque no siempre la he podido seguir al pie de la letra ya que iba pensando en diferentes formas de aplicarlo o en mejoras que surgían. Algunos cambios que han ido surgiendo es por ejemplo el foro o cambios en la interfaz, que le daban más dinamismo a la página.

Aun así ha habido algunas dificultades, sobre todo en la parte del minijuego y en cómo organizar los datos y cómo mostrarlos. También surgieron dificultades a la hora de mantener el progreso del usuario y conservar el estado de los niveles completados al navegar entre ellos. A pesar de esto considero que he avanzado sin grandes problemas.

### **6.4 Visión de futuro**

El proyecto tiene muchas posibilidades de mejorarlo en el futuro, ya que a día de hoy es una web funcional pero hay muchos detalles que la pueden hacer mucho más completa.

Una de las principales mejoras es añadir más minijuegos. Al principio del proyecto mi idea era crear varios minijuegos en vez de solo uno, como adivinar habilidades, adivinar el killer según sus estadísticas... También se podrían añadir estadísticas del jugador, como porcentaje de aciertos, número de intentos o clasificación entre usuarios. No he podido llevarlo a cabo porque quería añadir más funciones en vez de que solo se centrarse en los minijuegos.

Otra funcionalidad que tengo en mente, es que hubiese un sistema de “me gusta” en el foro, o que se pudieran organizar los temas como si fuese una función de preguntas y respuestas. Otra mejora interesante sería añadir perfiles públicos donde los usuarios puedan mostrar su progreso y actividad dentro de la web.

La funcionalidad que más importante me parece es añadir un panel de administración, donde los usuarios que tengan el rol de administrador puedan editar la información de la página o añadirla sin tener que acudir a la base de datos.

Esas son algunas de las funciones que se podría llegar a implementar, ya que tiene una base muy buena para que se mejore y convertir la web en una mucho más completa y profesional.

# 7. Glosario

## **API REST:**

Sistema que permite la comunicación entre el frontend y el backend a través de peticiones HTTP para intercambiar datos.

## **Backend:**

Parte del proyecto desarrollada con Spring Boot que se encarga de gestionar la lógica de la web, procesar datos, y comunicarse con la base de datos.

## **Base de datos:**

Sistema donde se almacena la información de la web, como los usuarios, personajes, objetos o comentarios. He usado PostgreSQL.

## **Cliente-servidor:**

Modelo en el que el frontend (cliente) solicita la información al backend (servidor) y responde con los datos necesarios.

## **Componente:**

Elemento reutilizable en el frontend en Vue, como por ejemplo el navbar que se usa en varias partes de la web.

## **Endpoint:**

Dirección de la API a la que se hacen peticiones, por ejemplo: /temas o /comentarios.

## **Frontend:**

Parte visual de la aplicación desarrollada con Vue que se encarga de mostrar la información y permitir la interacción del usuario.

## **HTTP:**

Protocolo utilizado para la comunicación entre cliente y servidor a través de peticiones web.

**JSON:**

Formato de datos utilizado para intercambiar información entre el frontend y backend.

**LocalStorage:**

Sistema de almacenamiento en el navegador que permite guardar datos del usuario.

**Navbar:**

Barra de navegación que permite moverse entre las diferentes secciones de la web.

**Persistencia de datos:**

Capacidad de guardar información para que no se pierda.

**Repository:**

Parte del backend que permite acceder a la base de datos de forma sencilla usando Spring Data JPA.

**Ruta:**

Dirección dentro del frontend que permite acceder a diferentes vistas de la web.

**Spring Boot:**

Framework de Java utilizado para desarrollar el backend y crear la API REST.

**Tomcat:**

Servidor web que utiliza Spring Boot para ejecutar la aplicación.

**Vue:**

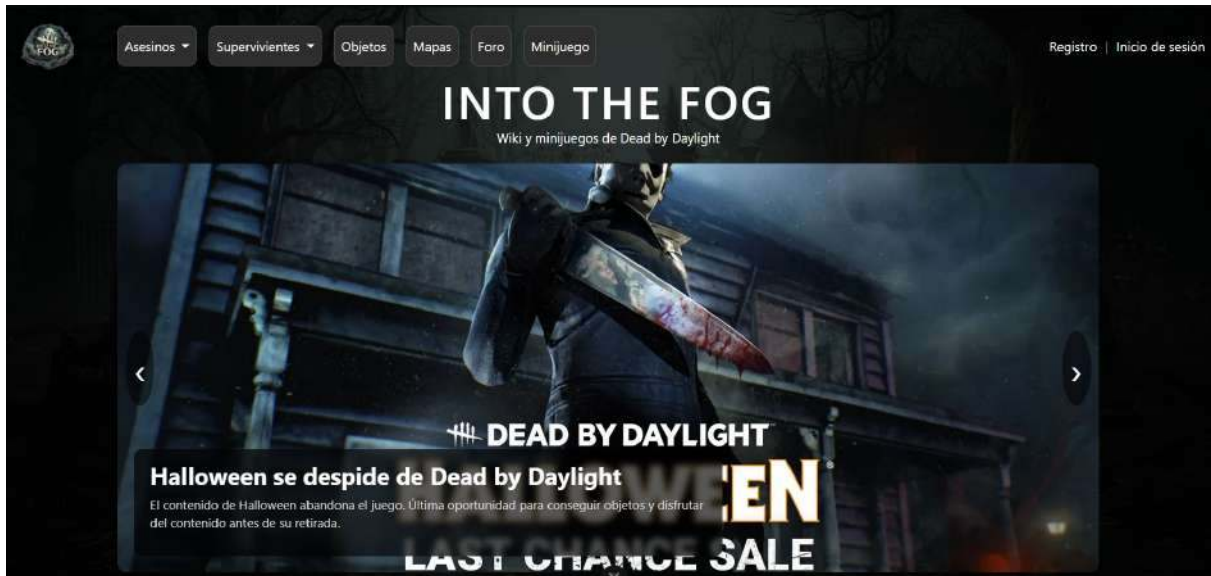
Framework de JavaScript utilizado para desarrollar el frontend de la web.

## 8. Bibliografía

- [BootstrapVue Documentation](#). (2024). *BootstrapVue*.
- [Spring Boot](#). (2025). *Spring Boot*.
- [Dead by Daylight Official Website](#). (2026). *Dead by Daylight*.
- [Dead by Daylight Wiki](#). (2026). *Fandom*.
- [DBDle](#). (2026). *DBDle*.
- [W3Schools Vue Tutorial](#). (2025). *W3Schools*.
- [Behaviour Interactive](#). (2026). *Behaviour Interactive*.

## 9. Anexos

Página inicio:



Vista asesinos:



## Vista asesino individual:

The screenshot shows the character profile for 'El Trampero' on the FOG website. The page features a dark, atmospheric background with a character portrait on the left. The character's name 'El Trampero' is prominently displayed with a heart icon, followed by the player's name 'Evan MacMillan'. Below this, four stat boxes provide details: 'Dificultad' (Fácil), 'Velocidad' (115% | 4.6 m/s), 'Arma' (Cuchillo de carnicero), and 'Poder' (Trampa para Osos). A 'Lore' section follows, containing a paragraph of text describing the character's nature as a man-monster with a mask and a hunting trap.

FOG Asesinos Supervivientes Objetos Mapas Foro Minijuego Registro Inicio de sesión

### El Trampero

Evan MacMillan

Dificultad	Velocidad	Arma	Poder
Fácil	115%   4.6 m/s	Cuchillo de carnicero	Trampa para Osos

#### Lore

Hay un gran monstruo de hombre, con una horrible sonrisa desgarrada en la máscara que acecha cada uno de mis movimientos. Al igual que un cazador, él nos rastrea, preparando trampas devastadoras entre la vegetación. La vigilancia extrema y un paso ligero es esencial para evitar el apretujamiento sangriento de una trampa para osos o las garras de lo que he llamado "El Asesino Sonriente Miserable". Muchas veces ha logrado atráparme en una de estas trampas. No tengo más que miedo por la sensación de las puntas de la trampa contundentes rechinando contra mis huesos o por los pasos pesados y deliberados que siempre inevitablemente revelan una sonrisa sádica enyesada. Con las trampas y el Trampero, hasta ahora ha sido una batalla constante entre mirar hacia arriba y mirar hacia abajo.

## Vista supervivientes:

The screenshot shows the 'Supervivientes' page on the FOG website. The page features a dark, atmospheric background with a grid of character portraits. Each portrait is accompanied by the character's name and their nationality. The characters listed are Dwight Fairfield (Americano), Meg Thomas (Americana), Claudette Morel (Canadiense), Jake Park (Coreano), William Bill Overbeck (Americano), Nea Karlsson (Sueca), David King, Laurie Rade, and Ace Visconti.

FOG Asesinos Supervivientes Objetos Mapas Foro Minijuego Registro Inicio de sesión

## Supervivientes

Dwight Fairfield Americano	Meg Thomas Americana	Claudette Morel Canadiense
Jake Park Coreano	William Bill Overbeck Americano	Nea Karlsson Sueca
David King	Laurie Rade	Ace Visconti

## Vista superviviente individual:

The screenshot shows a user profile for Dwight Fairfield. At the top, there is a navigation bar with a logo and menu items: Asesinos, Supervivientes, Objetos, Mapas, Foro, and Minijuego. On the right, there are links for Registro and Inicio de sesión. The profile features a portrait of Dwight Fairfield, his name with a heart icon, and three information boxes: Nacionalidad (Americano), Género (Masculino), and Fecha (2016-06-14). Below this is a 'Lore' section with a paragraph of text.

**Dwight Fairfield** ❤️

Nacionalidad: Americano

Género: Masculino

Fecha: 2016-06-14

**Lore**

Dwight era un chico tímido y delgado en el instituto. Siempre había deseado ser popular, pero carecía del carisma necesario. Intentó entrar en el equipo de fútbol americano, pero no lo seleccionaron; el equipo de baloncesto ni se fijó en él, y sus notas estaban bastante por debajo de la media. Un fin de semana, durante una actividad para fomentar el espíritu de equipo en su trabajo de mala muerte, el jefe de Dwight se los llevó a un bosque y les dio a probar licor casero de su receta familiar. Solo recordaba haber tomado un trago cuando se despertó al día siguiente, completamente solo.

## Vista objetos:

The screenshot shows an 'Objetos' (Items) page. It has the same navigation bar as the previous page. The page title is 'Objetos'. Below the title, there are six item cards, each with an icon, a title, and a description of its effects.

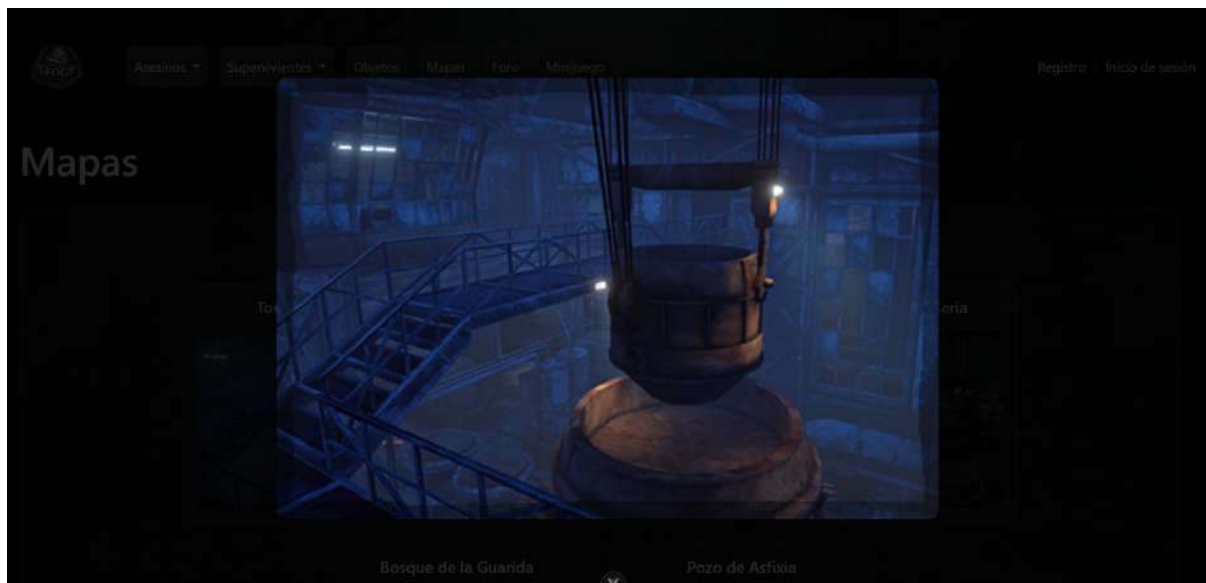
**Objetos**

- Botiquín guardabosques**: Con 24 cargas, aumenta la velocidad de curación a otros un 50%, permite usar Autocuración y reduce la velocidad y eficiencia al curarte un 33%.
- Caja de herramientas de Alex**: Con 18 cargas, aumenta la velocidad de reparación un 10%, permite usar Sabotaje y Aumenta la velocidad de sabotaje un 100%.
- Caja de herramientas de ingeniería**: Con 16 cargas, Aumenta la velocidad de reparación un 100%, permite usar Sabotaje y aumenta la velocidad de sabotaje un 10%.
- Linterna multiusos**: Con 12 segundos de uso, aumenta el brillo del haz de la linterna un 30%, Aumenta la duración de Ceguera un 15% y reduce la precisión del haz de la linterna un 20%.
- Botiquín de emergencia**: Con 24 cargas, aumenta la velocidad a la que curas a otros un 45%, permite usar Autocuración y reduce la velocidad y eficiencia la curarte un 33%.
- Caja de herramientas de mecánica**: Con 16 cargas, aumenta la velocidad de reparación un 70%, permite usar Sabotaje y aumenta la velocidad de Sabotaje un 25%.

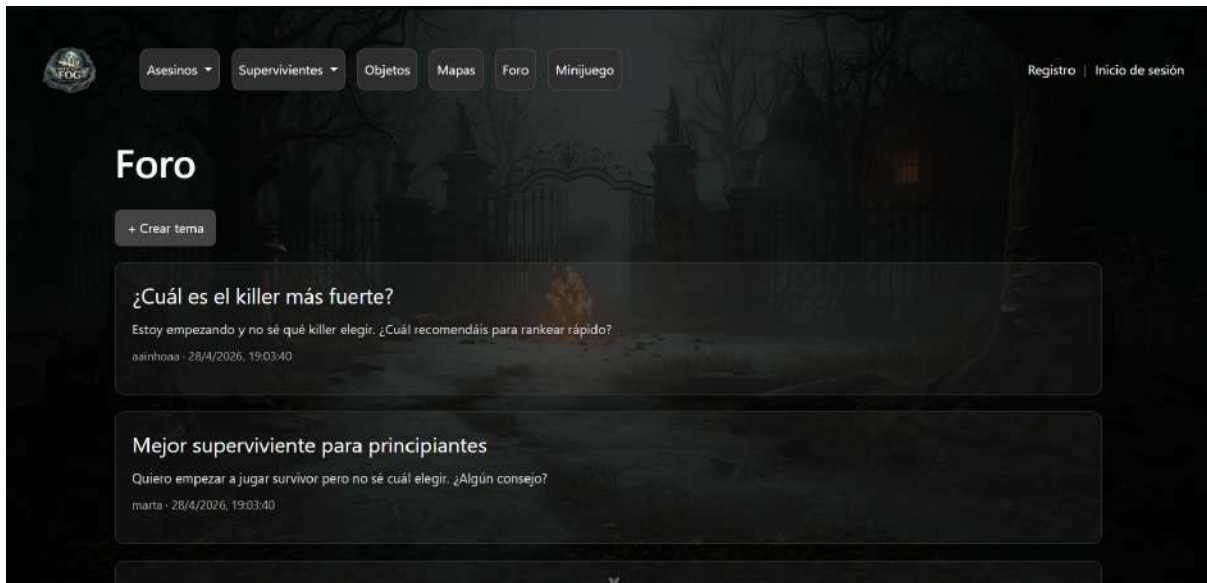
Vista mapas:



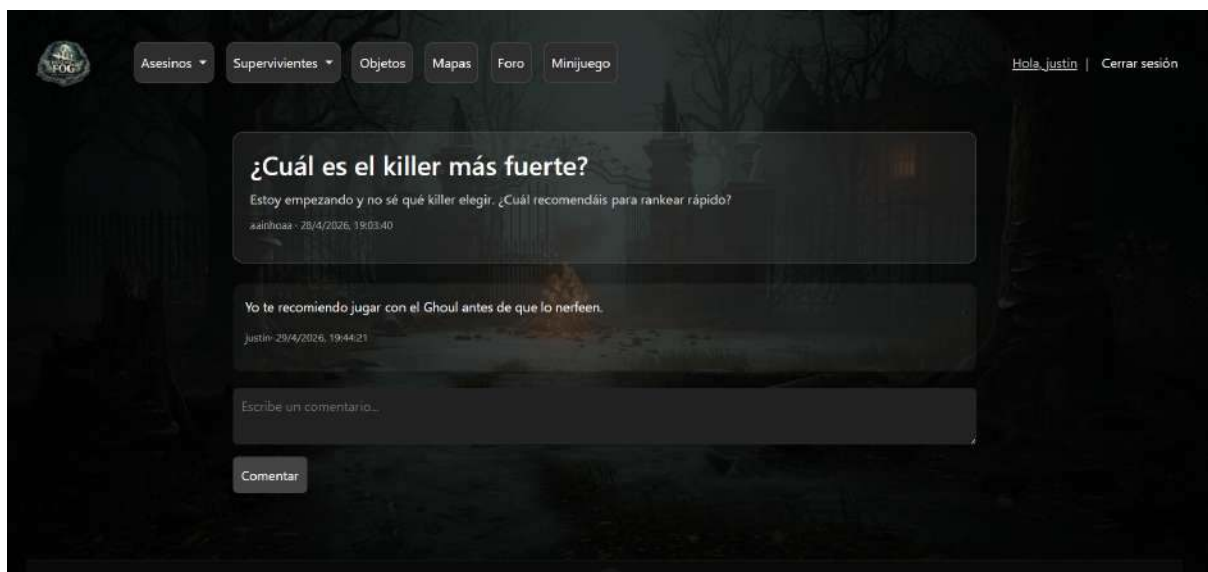
Imagen mapa (modal):



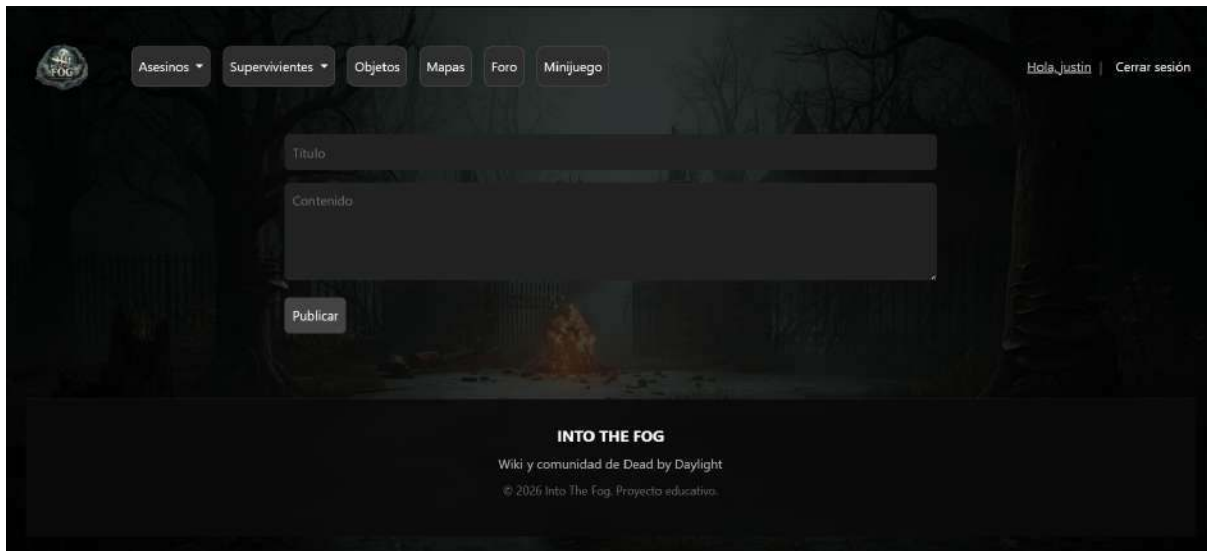
## Foro:



## Comentarios:



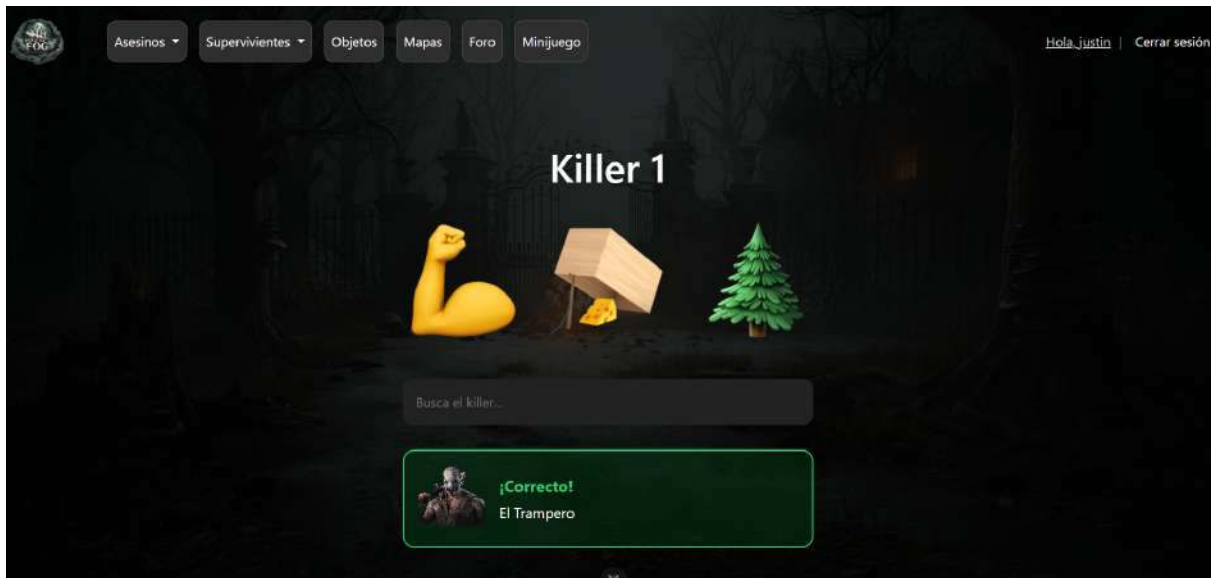
Añadir tema:



Minijuegos:



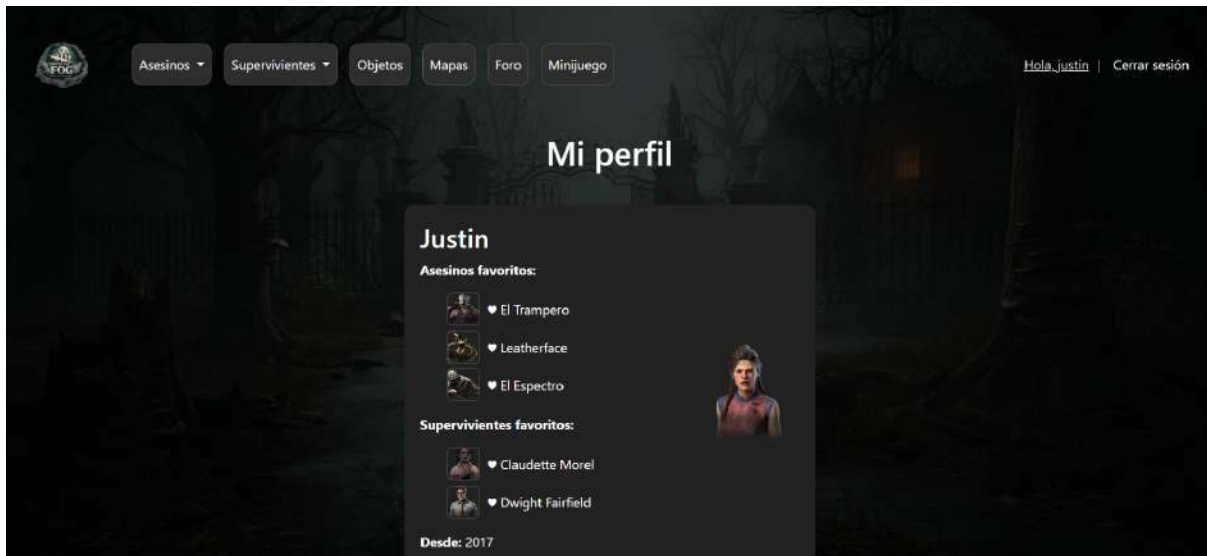
Minijuego acertado:



Minijuego en blanco:



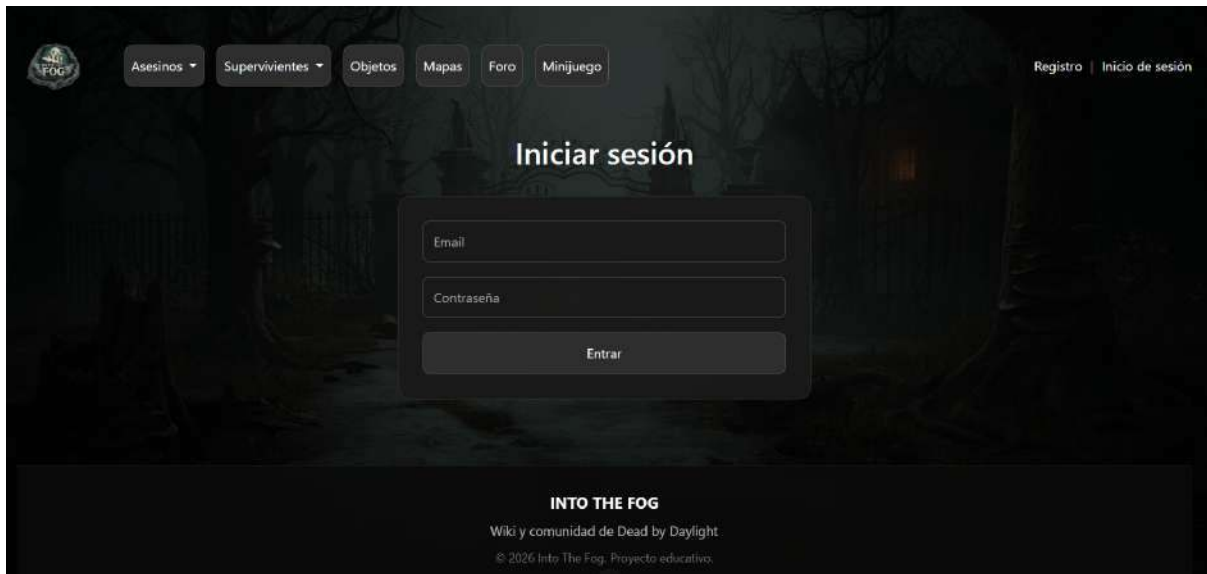
Perfil:



Registro de usuario:



Iniciar sesión:



Perfil al registrarse:



## Nota sobre el uso de inteligencia artificial

Durante el desarrollo del proyecto he utilizado herramientas de inteligencia artificial como Chat GPT para ayudarme tanto en el frontend como en el backend de la aplicación. La IA me ha servido para resolver dudas de programación, corregir errores, mejorar partes del código y entender mejor tecnologías como Vue.js, Spring Boot y PostgreSQL.

En el frontend, he utilizado la IA para mejorar el diseño de la interfaz, adaptar componentes responsive, solucionar problemas de conexión con la API y optimizar algunas partes visuales de la web. En el backend, la IA me ha ayudado especialmente para hacer la conexión con la base de datos PostgreSQL y para resolver errores relacionados con Railway, Supabase y el despliegue de la aplicación.

Además, también he utilizado la IA como apoyo para redactar partes de la documentación y mejorar la presentación general del proyecto. Aun así, toda la integración, adaptación del código, pruebas y decisiones finales del proyecto han sido realizadas y revisadas manualmente por mí para asegurar el correcto funcionamiento de la aplicación.

## Licencia



[Licencia: CC BY-NC-ND 3.0 ES](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)