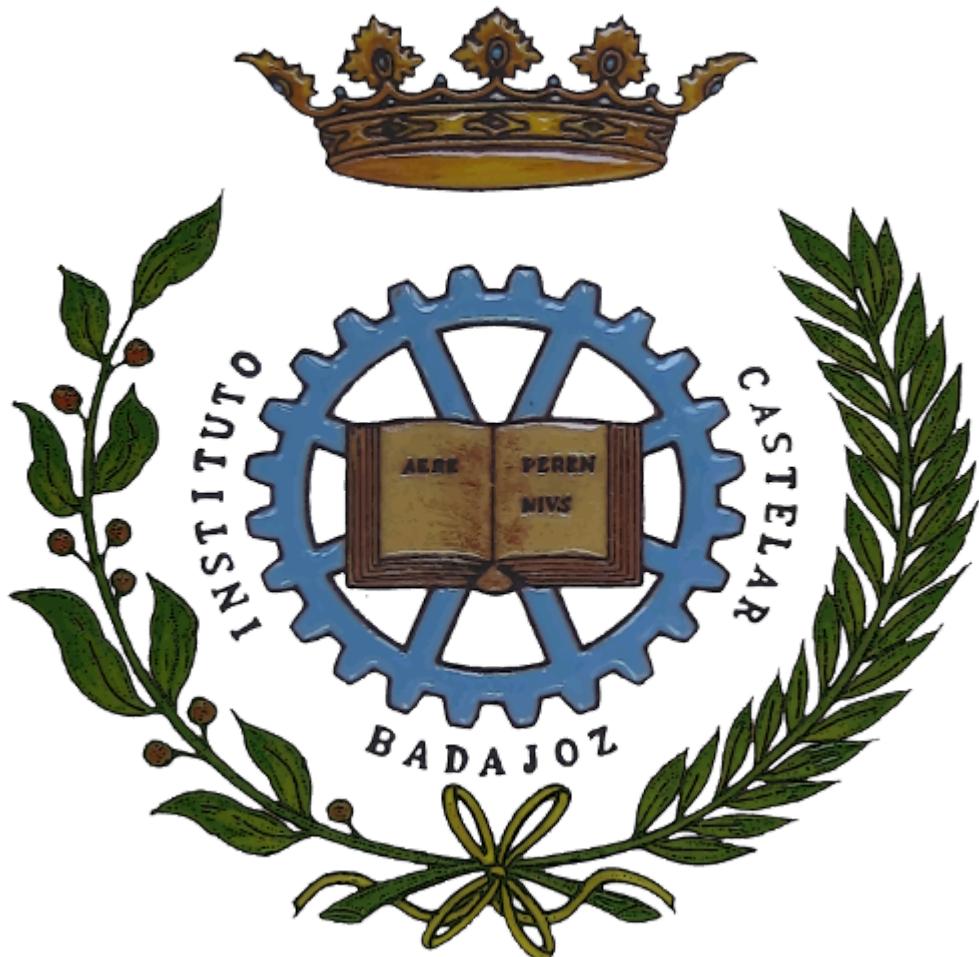


Proyecto Intermodular ASIR 2

Autor: Javier Romero



CREADOR DE FICHAS DE PERSONAJES

Proyecto Intermodular

Autor: Javier Romero

Tutor: Ignacio Segura

Fecha: Curso 2025/2026

Proyecto Intermodular ASIR 2

Autor: Javier Romero

ÍNDICE

ÍNDICE.....	2
1. INTRODUCCIÓN.....	4
Resumen.....	4
Contexto.....	5
Licencias.....	6
Política de privacidad.....	7
Licencias de contenido.....	7
Estado del Arte.....	8
2. OBJETIVOS.....	9
3. ESTUDIO DE LA VIABILIDAD.....	10
Situación Inicial.....	10
¿Qué funcionalidades necesita nuestra aplicación?.....	10
Funcionalidades Básicas:.....	10
Funcionalidades Adicionales.....	11
¿Qué es viable?.....	12
Requisitos del Sistema.....	12
4. ANÁLISIS DEL SISTEMA.....	14
Estructura General.....	14
FrontEnd.....	15
Formulario de Inicio de Sesión y Registro.....	16
Rutas Protegidas.....	17
Panel de Usuario.....	18
Panel Compartido.....	20
Formulario de Creación de personajes.....	21
Backend.....	22
Autentificación de Usuarios.....	22
API.....	23
Conexión con la Base de Datos.....	23
5. Diseño del Sistema.....	25
Base de Datos.....	25
Clases.....	25
Modelos.....	26
Clase Usuarios.....	26
Clase Personaje.....	26
Serializadores.....	27
Clase UsuarioSerializer.....	27
Clase PersonajeSerializer.....	27

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Tablas de la Base de Datos.....	28
Tabla Usuarios.....	28
Tabla Personajes.....	28
Endpoints de la API.....	29
Anexo 1: Terminología.....	30

Proyecto Intermodular ASIR 2

Autor: Javier Romero

1. INTRODUCCIÓN

Resumen

Este proyecto tiene como objetivo crear una página web full stacks en la cuál el usuario pueda crear y compartir personajes de un juego de rol de mesa con el resto de usuarios de manera sencilla e intuitiva.

La idea viene de que yo como Game Master¹ he observado que mostrar ejemplos de cómo se crea un personaje es la mejor manera de atraer nuevos jugadores a un juego de rol poco conocido. El personaje que el jugador va a personificar durante la campaña² es lo único que le importa a un jugador promedio, lo que hace que la creación del personaje sea la mayor barrera de entrada a un sistema nuevo.

Por ello, para una persona que va a lanzar un nuevo juego de rol es muy conveniente tener una manera sencilla de mostrar a los potenciales jugadores su sistema. Además, esto facilita el proceso de creación de personajes al ofrecer una manera interactiva a la clásica de leer el libro y llenar la ficha con papel y lápiz.

Gracias al auge de las mesas virtuales como Foundry VTT³ y Roll20, también se puede estudiar la posibilidad de ofrecer una integración entre estas mesas virtuales y nuestro producto. Por ejemplo, como Foundry ya da a la comunidad la oportunidad de crear plugins para ofrecer más funcionalidades, se podría intentar que desde nuestra página web se pueda generar un JSON o XML con el cual poder “importar” las fichas de personaje a Foundry.

Este mismo producto se podría vender a distintos diseñadores por separado. Cada juego tendría su propia página web independiente. No se podría juntar todo en una misma página ya que para eso se tendrían que poner de acuerdo todos los creadores y al final acabaría suponiendo en problemas. Además, si se mantiene por separado es más fácil adaptarse a las licencias de creación de contenido de terceros de cada uno.

Para este proyecto, se va a utilizar como ejemplo los sistemas Mothership de Tuesday Knight Games y Mythic Bastionland de Bastionland Press ya que sus licencias de terceros son mucho más permisivas.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Contexto

Como el cliente objetivo no es una empresa, sino individuos específicos dentro de un nicho, me parece adecuado empezar con una puesta en situación para que lo que se explique a continuación tenga sentido para cualquier persona que vea el proyecto.

Los interesados en este producto son diseñadores de juegos de rol de mesa independientes. En un mercado dominado por un titán mediático como lo es Dungeons and Dragons, es muy complicado darse a conocer. Los jugadores suelen ser muy reacios a probar cosas nuevas al estar acostumbrados a jugar siempre lo mismo.

¿Cómo pueden solventar esto los diseñadores? Dándoles la oportunidad con la creación de personajes de su sistema de manera fácil y gratuita. Ahí es donde entraría este producto. Ofrecería a los diseñadores el hacer una página web destinada a que los jugadores experimenten creando personajes antes de que den el paso a comprar el libro.

La creación de personajes es la parte más importante y llamativa para tu jugador promedio, lo que ayudaría inmensamente a los creadores a dar a conocer y vender su juego. La popularidad de herramientas similares, como [PathBuilder](#) para Pathfinder 2nd Edition o [COMP/CON](#) de Lancer RPG, demuestran que este tipo de aplicaciones son algo que potencia la visibilidad de dichos sistemas de juego y, en consecuencia, aumentando sus ventas.

Lo ideal sería que la aplicación mantenga un mismo esqueleto que poder aplicar a diferentes sistemas de juego de manera más sencilla y a partir de este personalizar el producto a cada cliente.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Licencias

LICENCIA DE SOFTWARE PROPIETARIO

Copyright © 2025

Todos los derechos reservados.

Se concede al usuario final una licencia limitada, no exclusiva, intransferible y revocable para acceder y utilizar la aplicación web Creador de Fichas de personajes únicamente para su uso personal, tras la compra o suscripción correspondiente.

Esta licencia no concede ningún derecho sobre el código fuente.

Queda expresamente prohibido al usuario:

- Copiar, reproducir, modificar o crear obras derivadas del software.
- Distribuir, revender, sublicenciar o poner la aplicación a disposición de terceros.
- Compartir cuentas, credenciales o accesos.
- Realizar ingeniería inversa, descompilación o desensamblado.
- Utilizar la aplicación para ofrecer servicios a terceros o con fines comerciales distintos al uso personal autorizado.

La aplicación, su código fuente, diseño, lógica, bases de datos y contenido son propiedad exclusiva del titular de los derechos y están protegidos por las leyes de propiedad intelectual aplicables.

El titular no garantiza la disponibilidad continua o ininterrumpida del servicio y se reserva el derecho de modificar, suspender o descontinuar la aplicación, total o parcialmente.

La aplicación se proporciona "TAL CUAL", sin garantías de ningún tipo, expresas o implícitas. En ningún caso el titular será responsable por daños directos, indirectos, incidentales o consecuentes derivados del uso de la aplicación.

Esta licencia se rescindirá automáticamente si el usuario incumple cualquiera de sus términos, sin derecho a reembolso.

Esta licencia se rige por las leyes de España.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Política de privacidad

El titular se compromete a cumplir con la normativa vigente en materia de protección de datos personales, en particular con el Reglamento (UE) 2016/679 (Reglamento General de Protección de Datos - RGPD) y la Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales (LOPD GDD).

Los datos personales proporcionados por los usuarios serán tratados conforme a lo dispuesto en la Política de Privacidad.

Licencias de contenido

El presente proyecto tiene carácter académico y no comercial. Los contenidos externos utilizados están amparados por licencias Creative Commons y se han incorporado conforme a los términos de uso académico permitidos por dichas licencias, respetando en todo momento la atribución de sus autores.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Estado del Arte

El sector de las tecnologías de la información mantiene un crecimiento sólido a nivel global. Las previsiones sitúan su expansión alrededor del 6 % anual, impulsada por la demanda de servicios, infraestructura, consultoría, migraciones a la nube y modernización de sistemas.

El cloud computing se consolida como arquitectura dominante. Los modelos híbridos y multicloud se expanden debido a la necesidad de flexibilidad, escalabilidad y control regulatorio. Paralelamente, el edge computing y el IoT ganan protagonismo para aplicaciones con baja latencia y procesamiento distribuido, especialmente en sectores industriales, ciudades inteligentes y logística.

La ciberseguridad se ha convertido en un eje central. El aumento de dispositivos conectados, la expansión del perímetro digital, las normativas de privacidad, entre otros obligan a integrar seguridad desde el diseño. La complejidad de las arquitecturas modernas exige nuevas prácticas de gobernanza, monitorización y gestión del riesgo.

El hardware y los semiconductores continúan siendo un pilar crítico. La demanda de centros de datos, sistemas edge y dispositivos IoT se enfrenta a tensiones en la cadena de suministro y a dependencias geopolíticas. La eficiencia energética y la sostenibilidad emergen como factores decisivos: los centros de datos y la infraestructura digital deben responder a restricciones ambientales y regulatorias.

A nivel organizativo, persisten desafíos importantes. La adopción tecnológica es desigual entre regiones y empresas, especialmente en pymes. La complejidad técnica aumenta y exige perfiles más especializados, lo que agrava la escasez de talento en áreas como arquitectura distribuida, seguridad y administración cloud. La dependencia de proveedores globales y la necesidad de soberanía digital generan debates sobre control, costes y resiliencia.

En los próximos años es previsible una consolidación del modelo basado en nube, edge y servicios distribuidos, acompañado de un énfasis creciente en sostenibilidad, seguridad y gobernanza. El avance del sector no depende exclusivamente de nuevas tecnologías, sino de la capacidad de las organizaciones para gestionarlas de forma eficaz, segura y económicamente viable.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

2. OBJETIVOS

- Haber llevado a cabo y comprendido las fases de realización de un proyecto informático basado en la web.
- Realizar una aplicación web empleando diferentes frameworks webs populares y demandados en el sector. En este caso serán Django y ReactJS.
- Implantación de la aplicación en la web mediante el uso de un hosting y el despliegue de un contenedor de Docker.
- Organización de un proyecto usando un sistema de control de versiones parecido al que se utilizaría en un proyecto real en una empresa.
- Conocer los casos de uso de diferente software y recursos entendiendo que licencias son necesarias para utilizarlos sin violar derechos de autor.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

3. ESTUDIO DE LA VIABILIDAD

Esta etapa del proyecto consiste en analizar qué tan factible es llevar a cabo cada uno de los objetivos y desarrollar cada una de las funcionalidades propuestas en este proyecto. Para ello, primero se tendrá que describir las necesidades básicas de nuestra aplicación e ir definiendo qué recursos y conocimientos se necesitarán para crearlas.

Situación Inicial

En nuestro caso, partimos de cero. Esta es una aplicación independiente que no necesita de ningún software o hardware de base.

¿Qué funcionalidades necesita nuestra aplicación?

Voy a dividir las funcionalidades del proyecto en dos tipos: funcionalidades básicas y funcionalidades adicionales. Las funcionalidades básicas son aquellas que nuestro proyecto necesita para funcionar de manera correcta y para cumplir con el objetivo principal del mismo, crear fichas de personaje para juegos de rol. Estas deberán ser creadas durante la fase de desarrollo sí o sí para que el proyecto pueda salir adelante.

Las funcionalidades adicionales son extras que harán que la aplicación sea más llamativa para posibles usuarios y clientes. El estudio de la viabilidad se va a centrar en estas funcionalidades, ya que son las que más van a dificultar el desarrollo del proyecto. Se llevarán a cabo durante la fase de desarrollo en función de la viabilidad de cada una y del tiempo que se disponga.

Funcionalidades Básicas:

Vamos a describir cada funcionalidad y explicar por qué son necesarias para nuestro proyecto.

- **Base de Datos:** necesitaremos una base de datos en la que guardar la información sobre los usuarios dados de alta en nuestra página web y sobre los personajes que haya creado cada uno. Sin una base de datos, nuestra aplicación sería muy básica y la mayoría de funcionalidades planeadas no se podrían implementar. Se va a emplear MySQL como sistema gestor de bases de datos por su facilidad de uso, a pesar de que, a la hora de guardar información sobre los personajes puede resultar un poco lioso.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

- **API:** es necesario crear una API para que nuestra aplicación pueda comunicarse con la base de datos de manera segura. Esté será el backend de nuestra aplicación, para el que usaremos la librería de django REST_Framework. Si la comunicación estuviera en el mismo sitio que la interfaz gráfica, esto supondría un problema de seguridad debido a la posible inyección de código.
- **Formularios:** Necesitamos un formulario para el inicio de sesión y registro de usuarios (se utilizará la misma base para los dos) y otro para la creación de personajes, que se adaptará al sistema pedido por el cliente. Sin formularios, no hay aplicación. Este será el frontend de la aplicación, que se encargará de recoger
- **Autentificación de usuarios:** un usuario deberá poder darse de alta en nuestra aplicación y poder iniciar sesión y acceder a sus personajes ya creados. Si no conseguimos hacer esto, un usuario no va a poder ver qué personajes ha creado, o los podría ver y modificar todos, o no vería nada.
- **Panel de Usuario:** en el panel un usuario podrá ver la información sobre su cuenta y podrá ver, editar, crear o borrar sus personajes. Este será el “hub” principal de la página y lo primero que verá el usuario cuando inicie sesión. Esta es otra funcionalidad esencial.

Funcionalidades Adicionales

Estos son extras que iremos añadiendo a la aplicación una vez se terminen las básicas

- **Panel compartido:** los usuarios podrán acceder a otro panel donde podrán ver personajes creados por otros usuarios. No todos los personajes se podrán ver, sólo aquellos que el usuario propietario haya decidido compartir (se añadirá la opción en el panel del usuario). Además, se podrán copiar los personajes para que se añadan en su panel y puedan editártos.
- **Animación de dados:** en el formulario de creación de personajes, se intentará añadir una animación que muestre una tirada de dados. El resultado que aparezca en la animación será reflejado en la ficha. Esto es algo muy llamativo para el usuario final, por experiencia como Game Master, los jugadores se vuelven locos por una animación de dados bien hecha.
- **Implementación en Foundry:** el usuario podrá exportar un archivo .json que se podrá importar en la aplicación [FoundryVTT](#). La idea sería crear un plugin que nos permita importar el json y de ahí sacar la información y generará una ficha en Foundry que refleje el personaje creado en nuestra aplicación. Esto es algo que haría que nuestra aplicación resalte sobre las demás.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

¿Qué es viable?

En principio, todas las funcionalidades básicas son algo asequibles de realizar con los conocimientos básicos sobre desarrollo web. La creación de APIs, conexión con una base de datos, operaciones de creación, actualización, borrado y lectura de datos en la base de datos mediante los formularios y el diseño de formularios e interfaces es lo único que se necesita para realizar estas funcionalidades. En la parte de diseño y desarrollo, se detallará el proceso y todo lo necesario. En cambio, en las funcionalidades adicionales es donde recae la mayor dificultad del proyecto.

El panel compartido es lo más sencillo de implementar, ya que es muy parecido a lo que será el panel de cada usuario. Lo más complicado sería que un usuario pueda copiar un personaje de otra persona a su panel, pero tampoco tiene mucha complejidad técnica.

La animación de los datos va a ser algo curioso de realizar. La animación como tal es algo sencillo de hacer con conocimientos básicos de modelado 3D, es más, seguro que los modelos de los datos se pueden sacar de repositorios de assets gratuitos. Lo difícil va a ser hacer que el resultado de la tirada de datos (una función de generación de números aleatorios) se corresponda con lo que aparece en la animación. Es algo que ya se ha hecho, así que es posible, pero va a ser complicado.

La implementación con Foundry es, en principio, es algo sencillo, es leer un json y generar una pestaña a partir de esos datos. El problema es que, como la mayoría de módulos de Foundry están hechos por miembros de la comunidad, cada uno es un mundo. Habría que leer el código escrito por otras personas, entenderlo y modificarlo. Si no se pudiera realizar, se intentaría hacer una simulación de cómo funcionaría en una página independiente.

Requisitos del Sistema

No solo tenemos que tener en cuenta las dificultades técnicas, sino que también hay que detallar lo que necesitamos para que nuestra aplicación funcione correctamente en la web.

- **Nombre de dominio:** sin él, los usuarios no serían capaces de acceder a la web. En una situación real, los costes del dominio estarían incluidos en el precio, ya que el cliente será el que elija el nombre de la aplicación. En el proyecto vamos a utilizar un nombre genérico y, si es posible, gratuito.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

- **Hosting:** nuestra página tendrá que estar alojada en algún servidor web. Esto también lo elegiría el cliente, pero para el proyecto habrá que escoger otro. Como nuestra aplicación es full stacks, tiene frontend, backend y una base de datos, será más complicado encontrar un servicio de hosting gratuito.
- **Licencia de MySQL:** aunque de cara al proyecto no haya que pagar nada ya que MySQL ofrece opciones gratuitas, habría que escoger una de las licencias comerciales que ofrece MySQL para aplicaciones en producción. En principio se escogería la versión estándar.
- **Licencias del Juego de Rol:** cada juego de rol tiene su propia licencia para la creación de contenido por terceros. En el caso real, cómo estaríamos trabajando directamente con los creadores, no haría falta. Sin embargo, en contexto del proyecto, sí que tendríamos que adherirnos a lo que digan las licencias de terceros, por lo que convendría escoger casos en los que la licencia sea permisiva.

Proyecto Intermodular ASIR 2

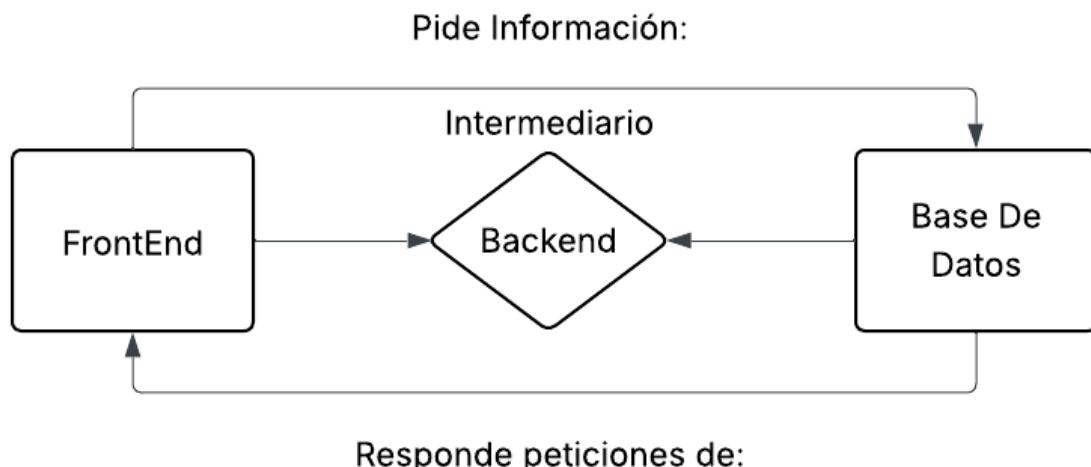
Autor: Javier Romero

4. ANÁLISIS DEL SISTEMA

Vamos a detallar las especificaciones de aquellas funcionalidades detalladas en apartados anteriores, tanto el apartado técnico como el gráfico.

Estructura General

Nuestra aplicación es full stacks, así que necesitamos que el frontend y el backend sean capaces de hablarse para que así la página pueda interactuar con la base de datos.



El backend hace de intermediario entre el frontend y la base de datos. Se podría prescindir de uno de ellos, pero eso haría el desarrollo de la parte faltante mucho más complicado. Hacer una interfaz gráfica es mucho más complicado en Django, que utiliza un sistema de plantillas HTML, y hacer la comunicación con la base de datos en React sería algo muy poco eficiente, ya que no tiene ni la mitad de herramientas con las que cuenta Django.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Respecto al funcionamiento, por ejemplo, digamos que el usuario quiere acceder a su panel de usuario. Primero, el frontend haría una petición HTTP al backend preguntándole por los personajes que le pertenecen a dicho usuario. Cuando llegue, el backend verificará que la petición es válida y, si lo es, hará una consulta a la base de datos. Después, el resultado de dicha consulta se le enviará como respuesta de la petición HTTP original al frontend junto con un código que indique que la petición ha sido exitosa. Esta explicación es un poco básica, en el código se realizará un ejemplo similar ya con la aplicación montada.

FrontEnd

Está es la parte de nuestro proyecto que verá el usuario final, y es donde se detallarán los requisitos y estructura de todos los formularios e interfaces de usuario que se van a implementar.

Para facilitar nuestro trabajo, vamos a utilizar un framework de JavaScript, React. Es probablemente uno de los más sencillos de aprender por la gran comunidad que se ha ido formando a lo largo de los años y por su popularidad de uso. En el mundo de las aplicaciones web existen dos tipos, las de nivel cliente y las de nivel empresarial.

Muy resumidamente, las de nivel cliente son las que podemos acceder desde nuestro navegador (puede acceder cualquiera), y las de nivel empresarial son las que utilizan las empresas de manera privada para sus propias necesidades.

React es el framework web que domina en el sector de las aplicaciones de nivel cliente, mientras que PHP, AngularJS o JSP son mucho más populares en el nivel empresarial.

Este framework nos va a ayudar a crear las interfaces gráficas gracias a que combina HTML y JavaScript en un lenguaje propio llamado jsx. Además contamos con todas las librerías y herramientas creadas por la comunidad, que nos permiten hacer cosas que con solo JavaScript serían muy complicadas de realizar.

¡OJO! En esta primera entrega de diciembre, toda las interfaces son maquetas sin funcionalidades para enseñar cómo se verían sin tener que desarrollar el framework. En las siguientes entregas se sustituirán por las versiones reales, junto con cualquier explicación de los cambios que se hayan tenido que hacer.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Formulario de Inicio de Sesión y Registro

Estos dos formularios se pueden meter en el mismo saco ya que se va a emplear el mismo como esqueleto, gracias a que React nos permite reutilizar componentes tantas veces como queramos. Para registrarse, un usuario deberá introducir un nombre de usuario único, un email que no se haya dado de alta en la página con anterioridad y una contraseña válida. Si los datos son correctos, el usuario estará registrado en la página y será redirigido a la página de inicio de sesión, donde bastará con que introduzca solo su nombre de usuario y contraseña.

Por la parte gráfica, el formulario serán dos cuadros de texto, uno de tipo contraseña para que al introducirla no se vea lo que escribe el usuario y un botón de enviar. Cuando se envíe el formulario, la información en los cuadros de texto se enviará al backend con una petición HTTP y ahí se validarán los datos. Si todos son correctos, el usuario se habrá dado de alta en nuestra página.

Este sería el formulario de registro:

Regístrate

Nombre de Usuario:

Email:

Contraseña:

La contraseña debe tener más de 8 caracteres y al menos un número

Enviar

¿Ya tienes cuenta? [Inicia Sesión](#)

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Este sería el formulario de inicio de sesión:

Inicia Sesión

Nombre de Usuario:

Contraseña:

Enviar

¿No tienes cuenta? [Regístrate](#)

Rutas Protegidas

Solo los usuarios dados de alta en nuestra página podrán ver el panel de usuario, el panel compartido y el formulario de creación de personajes. Para ello utilizaremos las funciones de redirección a otras páginas que ofrece React. Si un usuario que no esté dado de alta o que no haya iniciado sesión en nuestra página intentase acceder a ellas, se le redirigirá a la página del formulario de registro.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Panel de Usuario

Aquí los usuarios registrados podrán ver información sobre su cuenta, nombre de usuario y correo electrónico. En la parte derecha, aparecerán los personajes creados por dicho usuario. Cada personaje tendrá su propia “tarjeta” donde aparecerá el nombre del personaje y una descripción proporcionada por el usuario. También tendrá un botón para editar el personaje que le redirigirá al formulario de creación de personajes, y otro para borrar el personaje, que levantará una pestaña de confirmación antes de borrar definitivamente.

The screenshot shows a user interface for a RPG application. At the top, there's a navigation bar with links: 'Mothership RPG', 'Mis Personajes', 'Compartido', and 'Cerrar Sesión'. Below this, on the left, is a sidebar containing the user's details: 'Nombre de Usuario: Ejemplo' and 'Email: ejemplo@ejemplo.com', with a 'Nuevo Personaje' button. The main area is titled 'Personajes Creados:' and lists two characters, each with an 'Editar', 'Exportar', and 'Eliminar' button. The characters listed are 'Nombre: Ejemplo'.

Este a su vez estará dividido en tres secciones

Mothership RPG Mis Personajes Compartido Cerrar Sesión

Esta es la barra de navegación, cada uno de los botones llevará a una página diferente. En el caso de “Cerrar Sesión”, este eliminará la sesión del usuario y le redirigirá al formulario de inicio de sesión.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Nombre de Usuario: Ejemplo

Email: ejemplo@ejemplo.com

[Nuevo Personaje](#)

Aquí mostraremos la información del usuario. Cada usuario verá su información si y sólo si está dado de alta y ha iniciado sesión y, por tanto, sus datos están guardados en la base de datos

Personajes Creados:

Nombre: Ejemplo Editar Exportar Eliminar	Nombre: Ejemplo Editar Exportar Eliminar
Nombre: Ejemplo Editar Exportar Eliminar	Nombre: Ejemplo Editar Exportar Eliminar

En esta sección aparecerán todos los personajes creados por el usuario ordenados por fecha de última edición, es decir, el último que haya editado será el primero que aparezca. Esta parte está preparada para mostrar un total de 8 personajes al mismo tiempo, cuando el número exceda 8, se guardarán en una segunda pestaña.

De cada personaje aparecerá su nombre y la fecha de última creación. Debajo, cada botón tendrá una función específica. El botón de editar llevará al usuario al formulario de creación de personajes, con los datos del personaje elegido ya apareciendo rellenos en el formulario. El botón de exportar generará un fichero en formato JSON con los datos del personaje. Por último, el botón de borrar eliminará al personaje de tanto la página como de la base de datos.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Panel Compartido

Este panel será muy parecido al anterior, solo que aparecerán personajes de otros usuarios en vez de solo los del usuario que esté conectado. La estructura de las tarjetas de cada usuario serán similares también, solo que tendrá un botón que en vez de editar, cree una copia de un personaje en el panel del usuario para que lo pueda editar.

The screenshot shows a dark header bar with navigation links: 'Mothership RPG', 'Mis Personajes', 'Compartido', and 'Cerrar Sesión'. Below the header, the title 'Personajes:' is centered above a search bar. The main content area displays two character cards. Each card contains a placeholder name ('Nombre: Ejemplo'), a blue 'Importar' button, and another placeholder name ('Nombre: Ejemplo') below it, also with a blue 'Importar' button. A green 'Buscar' button is located to the right of the search bar.

El panel compartido mostrará los personajes de manera similar a como lo hace el panel de usuario. El nav es el mismo en todas las páginas.

Personajes:

Tiene una barra de búsqueda donde buscar a los personajes por el nombre que haya puesto su usuario correspondiente. La idea es que cada vez que se escriba una letra, la búsqueda se actualice automáticamente. Esto se mostrará en la parte de desarrollo.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Nombre: Ejemplo	Nombre: Ejemplo
<button>Importar</button>	<button>Importar</button>
Nombre: Ejemplo	Nombre: Ejemplo
<button>Importar</button>	<button>Importar</button>

En este panel, cada personaje mostrará el usuario que lo haya creado en vez de la fecha de modificación y el botón permitirá al usuario guardar una copia del personaje en su panel de usuario, para que le permita modificarlo.

Formulario de Creación de personajes

Durante esta fase del proyecto se va a mostrar una versión preliminar de este formulario, cuando se llegue a la fase de desarrollo se actualizará para que tenga su forma definitiva.

Este será el formulario de creación de personaje, por ahora solo vamos a mostrar lo esencial para hacernos una idea y para testear el formulario antes de pasar al formulario completo

Crea tu Personaje

Personal Details
Character Name

Pronouns

Roll your stats


Strength


Speed


Intellect


Combat

Roll your saves


Body


Fear


Sanity

Quieres que este personaje se comparta con el resto de usuarios?

Guardar

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Personal Details

Character Name

Pronouns

En esta sección del formulario el usuario introducirá el nombre de su personaje y los pronombres que correspondan a la identidad de género del mismo.

Roll your stats



Strength



Speed



Intellect



Combat

Roll your saves



Body



Fear



Sanity

Aquí el usuario podrá introducir las estadísticas de su personaje. Podrá hacerlo a mano o tendrá la opción de generarlos aleatoriamente si el usuario así lo desea mediante un botón que simule una tirada de dados

Proyecto Intermodular ASIR 2

Autor: Javier Romero

¿Quieres que este personaje se comparta con el resto de usuarios?

Guardar

Por último, si el usuario marca la casilla de verificación, el personaje que acaba de crear será visible para el resto de usuarios. Cuando le de al botón de guardar, le mostrará un mensaje de confirmación y se le redirigirá a su panel de usuario.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Backend

Esta es la manera más sencilla de hacer operaciones en la base de datos. Vamos a crear una API sencilla que nos permita comunicarnos tanto con el frontend como con la base de datos. Además esto añade una capa extra de seguridad a nuestra página, ya que un usuario no va a poder interactuar directamente con la base de datos, lo que evita problemas de inyección de código. Esta parte de la aplicación solo será visible para los desarrolladores.

Django es un framework backend de Python conocido como “batteries included”, es decir, incluye muchas herramientas y funcionalidades ya incluidas en el propio framework y que no hace falta desarrollarlas por nuestra cuenta. Por ejemplo, cuenta con un sistema de autenticación de usuarios propio y fácil de modificar para que se adapte a cualquier tipo de aplicación o con una plataforma API que nos ahorra el tener que emplear aplicaciones como Postman o Thunder Client para testear la API antes de que tengamos el frontend montado y conectado correctamente.

Autentificación de Usuarios

Tenemos que ser capaces de hacer que un usuario se pueda dar de alta en nuestra página, lo que nos permite que todo lo que haga se guarde en la base de datos y esté asociado a su propio usuario para que, por ejemplo, solo él pueda editar o borrar personajes que le pertenezcan. Vamos a modificar la clase que utiliza Django por defecto para que también nos deje añadir el email de nuestro usuario. Las especificaciones de las clases se añadirán en la sección de diseño.

Para la autenticación vamos a utilizar JSON Web Tokens para demostrar que un usuario está dado de alta en el sistema. Si el formulario de inicio de sesión es correcto, al usuario se le van a asociar dos cadenas encriptadas, el token de acceso y el token de refresco. La página verificará que el token de acceso es válido, y si lo es, dejará que el usuario acceda a la página. El token de refresco sirve para que el usuario pueda estar conectado a la página todo el tiempo que nosotros queramos. El token de acceso caduca en unos 10 minutos, cuando ese tiempo pase, se comprobará que el token de refresco es válido y si es correcto, le otorgará otro token de acceso al usuario. Cuando el token de refresco expire, el usuario tendrá que volver a iniciar sesión en la página.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

API

Gracias a ella nos podremos comunicar con la base de datos. Estará constituida por una serie de endpoints que estarán representados por rutas en nuestra página web. Cada endpoint estará asociado a una operación simple que realizaremos en la base de datos.

Para los usuarios, necesitaremos mandar información para que la escriba en la base de datos cuando este se registre y otra donde verificar si la información es correcta cuando intente iniciar sesión.

Para la creación de personajes, vamos a seguir el estándar de operaciones CRUD (Crear, Leer, Actualizar y Borrar). Cada una de estas tendrá una un endpoint propio para que las rutas de la API estén mejor organizadas. En el caso de que tuviésemos otra tabla, esta necesitaría sus endpoints CRUD independientes.

Además, necesitaremos un serializador que traduzca los datos que introducimos en los formularios a json para mandar la información al frontend (este tipo de peticiones HTTP se comunican en formato json).

Django viene con una herramienta que nos permite poder testear la API sin tener que tener el frontend montado completamente. Normalmente habría que utilizar una herramienta de terceros como Postman o Thunder Client para esto. Esto es importante para que luego no nos encontremos con errores durante el desarrollo del frontend que tengan que ver con el backend.

Conexión con la Base de Datos

Django utiliza lo que se conoce como ORM (Object Relational Mapping). Su función es coger una clase creada en Python y, a partir de esta, crear una tabla correspondiente en la base de datos. Nosotros necesitaremos crear una tabla para los usuarios y otra para los personajes. Además Django tendrá que crear sus propias tablas por temas de configuración.

Este ORM funciona independientemente del sistema de bases de datos que utilicemos gracias a que Django maneja las migraciones entre bases de datos por su propia cuenta, los desarrolladores solo tenemos que tener cuidado con seguir los estándares de creación de modelos que establece el framework.

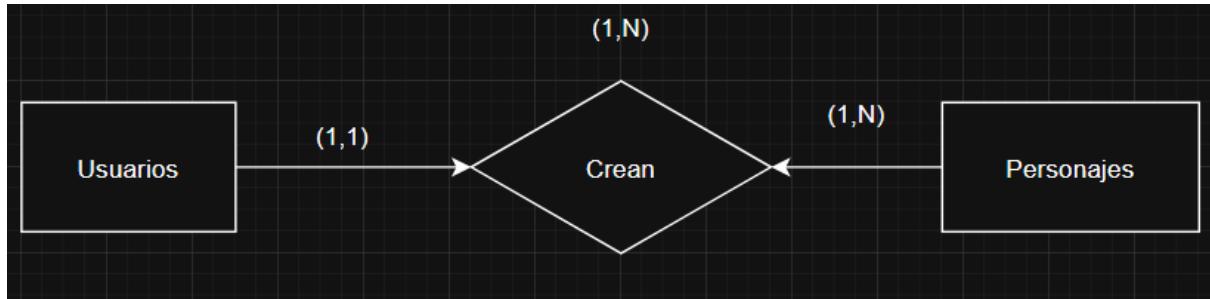
Proyecto Intermodular ASIR 2

Autor: Javier Romero

5. Diseño del Sistema

OJO! Esto es lo que se planifica que será necesario en esta fase del proyecto. Si se cambiase o añadiese algo durante la fase de desarrollo se detallará en un apartado extra.

Base de Datos



La base de datos solo va a necesitar de dos tablas, una donde se guardará la información sobre los usuarios que se registren en la página y otra para los personajes que cree cada usuario. Un usuario podrá crear uno o varios personajes, mientras que un personaje solo podrá ser creado por un único usuario, lo que significa que son una relación (1,N) y se tendría que transmitir la clave primaria de la tabla Usuarios como un campo en la tabla Personajes. Los atributos y campos de cada tabla se detallarán en apartados siguientes.

Clases

En este apartado se detallarán las clases necesarias para el correcto funcionamiento de nuestra aplicación. En nuestro caso, habrá dos tipos de clases, las que estén relacionadas con la base de datos, que se suelen denominar “modelos” y los serializadores.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Modelos

Como Django utiliza un sistema ORM para el manejo de base de datos, no es necesario que creamos directamente ninguna tabla en la base de datos. Tendremos que crear una tabla que defina la estructura, clave primaria, relaciones, tipos de campo, etc. para que el sistema de migraciones de Django las cree por nosotros.

OJO! No hace falta que creamos un Id para las clases, ya lo hace Django por defecto.

Clase Usuarios

Usuario	
Nombre:	str
Email:	str
Contrasena:	str
+__str__()	

Cada usuario deberá introducir su nombre de usuario, email y contraseña cuando se quiera registrar en nuestra página. Al iniciar sesión, si existe un usuario con el nombre de usuario introducido y los hashes de la contraseña coinciden, el usuario podrá acceder al resto de la página.

Clase Personaje

Personaje	
Nombre:	str
Clase:	str
Fecha_Creacion:	str
Creador:	str
+__str__()	

OJO! Esta clase es un placeholder. Representa a un personaje genérico sin especificar el sistema de juego que a elegir. Cuando la aplicación funcione con este molde, se aplicará a un sistema real.

Lo que importa extraer de este molde que estará presente en cualquier sistema son los campos Nombre (para identificar al personaje), Fecha_Creacion (para ordenarlos en función de cuando se han creado) y Creador (la clave foránea que representa la relación entre personaje y usuario).

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Serializadores

En una petición HTTP, necesitamos que la información viaje en formato JSON para que funcione correctamente. Un serializador se encarga de traducir la información de un registro en nuestra base de datos a JSON para poder mandarla en una petición HTTP. Por regla general, cada modelo necesita un serializador propio.

Clase UsuarioSerializer

UsuarioSerializer		
Model:	obj	
Fields:	list	
Extra_Kwargs:	dict	

Cada serializador tiene una estructura similar en cuanto a estructura de clase se refiere, lo que se especifica en cada atributo es lo que diferencia uno de otro.

En Model tenemos que poner que modelo va a serializar esta clase, por ejemplo, el modelo Usuario.

En Fields tenemos que poner los nombres de los campos/atributos de nuestro modelo, en el caso del usuario serán el Id (aquí sí que hay que especificarlo), Nombre, Email y Contraseña.

En Extra_Kwargs tenemos que poner configuraciones adicionales para hacer un mejor control de la información, por ejemplo, el campo Contraseña deberá ser de solo escritura para que no se mande el hash en la petición HTTP.

Clase PersonajeSerializer

PersonajeSerializer		
Model:	obj	
Fields:	list	
Extra_Kwargs:	dict	

Hay que tener en cuenta lo mismo que en el otro serializador, solo que cambiando los campos por los que correspondan en la clase Personaje.

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Tablas de la Base de Datos

Ahora que sabemos cómo se van a crear las tablas de nuestra base de datos, falta explicar qué va a aparecer exactamente en nuestras tablas.

Tabla Usuarios

Nombre del Campo	Tipo de Dato	¿Nulo?	Restricciones
Id	varchar	No	Clave Primaria
Nombre	varchar	No	
Email	varchar	No	
Contraseña	varchar	No	

Tabla Personajes

Nombre del Campo	Tipo de Dato	¿Nulo?	Restricciones
Id	varchar	No	Clave Primaria
Nombre	varchar	No	
Clase	varchar	Sí	
Fecha_Creacion	date	No	
Creador	varchar	No	Clave Foránea

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Endpoints de la API

```
.  
└── 127.0.0.1/  
    ├── api/  
    │   ├── user/  
    │   │   └── register  
    │   ├── token/  
    │   │   └── refresh  
    │   └── personaje/  
    │       └── delete  
    ├── admin  
    └── api-auth
```

Los endpoints se pueden representar como una estructura de carpetas o rutas de directorios, la raíz siendo la IP de nuestra aplicación (por ahora 127.0.0.1). Cada endpoint tendrá su propia función

- api/user/register/ será donde se creen los usuarios de nuestra aplicación
- api/token/ es donde se generarán los tokens de acceso y refresco
- api/token/refresh/ es donde usaremos el token de refresco para generar un token de acceso nuevo.
- api/personaje/ es donde se crearán los personajes
- api/personaje/delete/ servirá para borrar personajes
- admin/ es el panel de administrador de Django, no va a ser importante para el proyecto pero debe de estar ahí
- api-auth/ es un endpoint que crea djangorestframework por defecto para manejar la autenticación de usuarios

Proyecto Intermodular ASIR 2

Autor: Javier Romero

Anexo 1: Terminología

Para que haya un mejor entendimiento de los términos empleados que estén relacionados al hobby de los juegos de rol, voy un anexo con definiciones de dicha terminología:

1. **Game Master:** Jugador que se encarga de narrar y crear la trama, escenarios y personajes de una partida de rol. Además, es quién hace de “árbitro” y decide que es lo que ocurre cuando se tiran los dados.
2. **Campaña:** como se denomina a una serie de partidas sucesivas de rol que siguen una misma trama y a unos mismos personajes.
3. **FoundryVTT:** aplicación que se utiliza para crear mesas virtuales para que un grupo de personas pueda jugar una partida de rol online. Es un programa que instala un servidor web hosteado en un hosting propio o en uno alquilado como Forge. Este programa permite que los usuarios creen plugins para añadir funcionalidades extra.