

INSTITUTO DE EDUCACIÓN SECUNDARIA JOSÉ PLANES

Departamento de Informática y Comunicaciones

Técnico Superior en Desarrollo de aplicaciones Web

C/ Maestro Pérez Abadía, 2

30100 Espinardo – Murcia

T. 968 834 605

30010577@murciaeduca.es

www.iesjoseplanes.es

Memoria del proyecto

Desarrollo de aplicaciones web

DNDPedia - Tu enciclopedia todo en uno de Dragones y Mazmorras

Autores:

Javier Alonso Paniagua

Javier Crespo Ortés

Profesor-coordinador:

Luis Miguel Fernández Costa

Murcia, junio de 2024



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional](#).

Queremos agradecer a todos los profesores por la formación que nos han brindado y a nuestro tutor de prácticas por ayudarnos con consejos sobre la elaboración del proyecto.

Contenido

1 Resumen extendido	1
2 Palabras clave	1
3 Introducción	1
4 Estado del arte/trabajos relacionados	2
5 Análisis de objetivos y metodología.	3
5.1 Objetivos	3
5.2 Metodología	3
6 Diseño y resolución del trabajo realizado	4
6.1 Diseño de la Base de Datos	4
6.2 Login, Control Login y Logout	10
6.3 Creación del Buscador con sus filtros, resultados y pantalla de detalles	12
6.4 Creación de Solicitudes con sus filtros adecuados	16
6.5 Creación de Revisiones con sus resultados	19
6.6 Creación de Ficha de Personaje con sus filtros adecuados	21
6.7 Diseño	24
6.8 Despliegue de la Aplicación	25
7 Presupuesto	25
8 Conclusiones y vías futuras	26
9 Bibliografía/Webgrafía.	27

1 Resumen extendido

DNDPedia es una aplicación web desarrollada con Symfony y Bootstrap, con el objetivo de ser una enciclopedia útil y accesible para el usuario, quien podrá buscar lo que necesite con nuestro potente sistema de filtros y ver los detalles de su búsqueda.

Además, ofrecemos la creación de un personaje a raíz de un par de formularios los cuales recogen tus datos y te crea una pantalla la cual se puede imprimir mediante la combinación de teclas ctrl+P y está hecha de manera que te autogenera todo lo suficiente para poder jugar una partida de Dragones y Mazmorras al instante con tus amigos.

Y finalmente también se pueden enviar solicitudes de cualquier apartado, es decir una incorporación de datos a la base de datos. Se te generará una plantilla como los detalles del buscador para que un administrador revise y pueda validar o no validar.

2 Palabras clave

DNDPedia, Dragones y Mazmorras, rol, enciclopedia, buscador, filtros, ficha, personajes, revisiones, Symfony, Bootstrap, CSS.

3 Introducción

La idea de DNDPedia es poder facilitar la accesibilidad de D&D 5e (*Dungeons & Dragons quinta edición*) a los usuarios. Hay mucho contenido por internet, pero no está adaptado para gente nueva ni organizado de manera accesible, así que hemos decidido simplificarlo para los más nuevos, para buscar información y crearte una ficha de personaje completa. Y para los más expertos podéis incluso enviar solicitudes revisadas por el administrador para validarlas o no.

La idea es usar Symfony debido a que nos pareció una herramienta muy útil y completa para tanto el backend como el frontend y también para toda la parte del login y su control, Symfony es capaz de poder realizarlo todo si se le da el uso correcto y con los conocimientos adecuados.

Para el diseño decidimos utilizar tanto la base que aporta Bootstrap como CSS propio para complementar, ya que Bootstrap otorga elementos que están muy bien hechos y sin tener que importar nada que no sea el Bootstrap CDN y decidimos combinarlo con CSS porque así podemos tener una mejor comprensión de lo que hemos estado haciendo ya que todo cambio hecho en la web ha sido realizado a mano por nosotros.

4 Estado del arte/trabajos relacionados

Nuestra página web es una inspiración de Nivel20, la página web oficial de Dragones y Mazmorras..

Nivel20 al igual que DNDPedia es capaz de hacerte una búsqueda de la información que prefieras, así como poder crearte una ficha de personaje. Sin embargo, teníamos la sensación de que los filtros no eran lo suficientemente potentes y aparte no estaba muy adaptado para gente nueva tanto la creación de personaje como la información dada así que esa ha sido nuestra idea desde un principio y la cuál creemos que es nuestro punto fuerte que se diferencia de la página en la que nos inspiramos.

Adicionalmente, nuestra filosofía es que cualquier usuario sea capaz de acceder al contenido sin necesidad de registrarse o tener que pasar horas buscando por contenido adicional escondido en rincones inhóspitos de la web. En nuestra web, cualquiera puede acceder en cualquier momento sin necesidad de registro.

Las imágenes son sacadas de un banco de imágenes de libre uso, la fuente de letra del proyecto está sacada de un banco de fuentes libre de uso llamado *Dafont*, los iconos están sacados de un banco de iconos de libre uso llamado *Icofont* y las imágenes utilizadas para los elementos están sacadas de un banco de imágenes de libre uso llamado *iStock*.

5 Análisis de objetivos y metodología.

5.1 Objetivos

Una vez elegimos el tipo de proyecto y lo que se haría, decidimos fijar los objetivos principales de este:

- Creación de una base de datos relacional adecuada.
- Login y Logout de administradores.
- Creación de un menú dinámico que cambia en función de si el administrador está logueado o no.
- Creación de los filtros de Buscador además del resultado mediante queries y sus pantallas de detalles.
- Creación del formulario de Solicituds el cuál se utilizará para enviar peticiones.
- Creación y edición de la Ficha de Personaje el cuál a partir de un formulario nos creará una ficha jugable e imprimible.
- Creación de las Revisiones las cuales un administrador puede validar o no validar las solicitudes mandadas por un usuario.

5.2 Metodología

Para poder llevar a cabo los siguientes objetivos se ha llevado a cabo la siguiente metodología de trabajo:

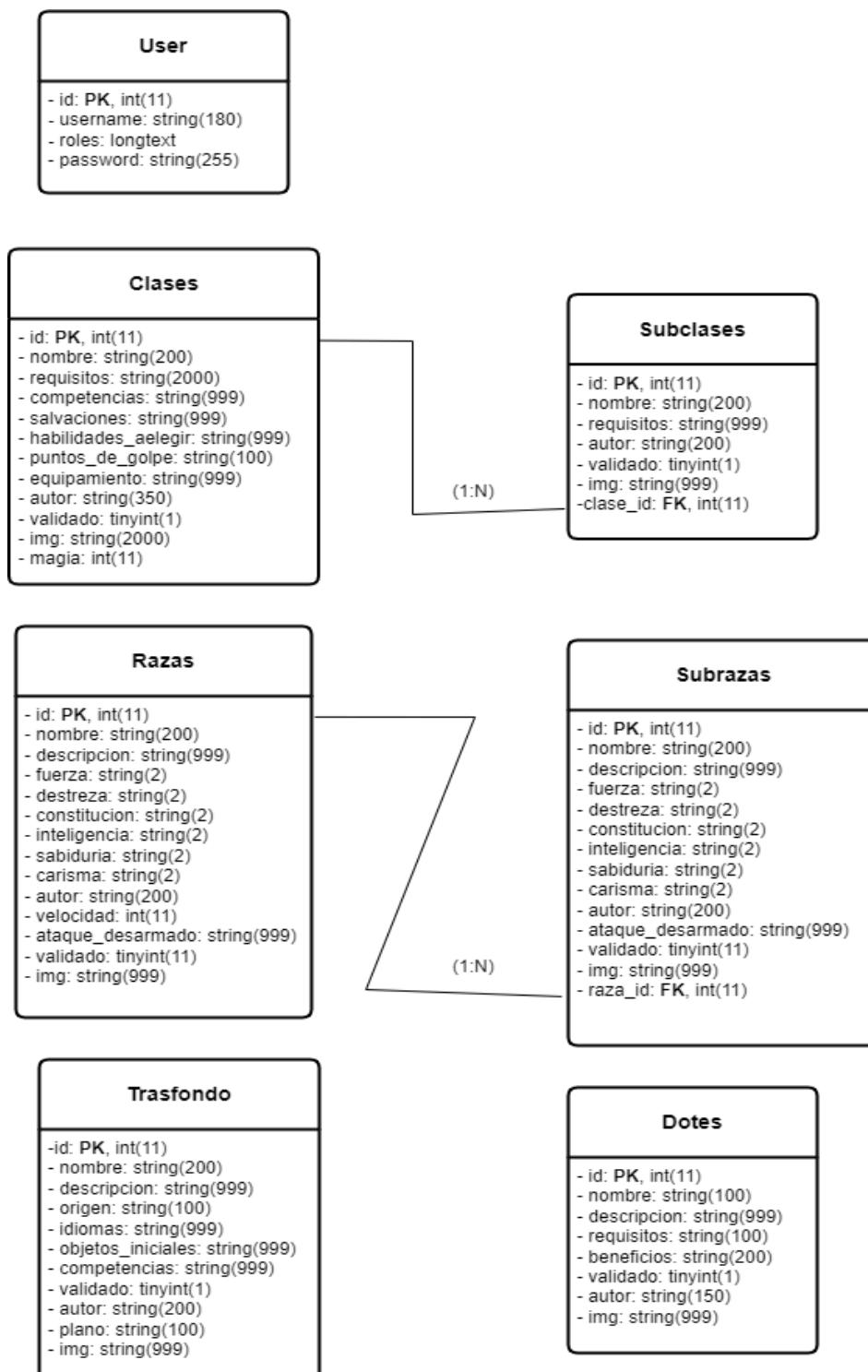
- Uso de Discord para realizar programación en pareja (*peer programming*) al menos una vez a la semana.
- Uso de GitHub para el control de versiones y para realizar tareas individuales que nos asignamos.
- Uso de Whatsapp para la comunicación por texto cuando no estábamos en llamada.
- Base de Datos del proyecto: La integrada por XAMPP, PHPMyAdmin, posteriormente exportada al servicio web homónimo proporcionado por Ionos.

6 Diseño y resolución del trabajo realizado

- Primero de todo vamos a exponer las principales partes de nuestro proyecto por separado. Estas partes son:
- Diseño de la Base de Datos
- Login, Control Login y Logout
- Creación del Buscador con sus filtros, resultados y pantallas de detalles.
- Creación de Solicituds con sus filtros adecuados.
- Creación de Revisiones con sus resultados.
- Creación de la ficha de Personaje con sus filtros adecuados.
- Diseño
- Despliegue de la Aplicación

6.1 Diseño de la Base de Datos

Primero decidimos crear la Base de Datos la cual sufrió modificaciones y adiciones, pero el resultado final fue el siguiente diagrama de relaciones de entidad:



Hechizos

- id: PK, int(11)
- nombre: string(200)
- descripcion: string(999)
- nivel: string(10)
- escuela: string(100)
- rango: int(11)
- componentes: string(999)
- validado: tinyint(1)
- autor: string(100)
- zona_efecto: string(200)
- tiempo_de_ejecucion: string(100)
- duracion: string(100)
- img: string(999)

Habilidades

- id: PK, int(11)
- nombre: string(200)
- descripcion: string(9999)
- autor: string(200)
- validado: tinyint(1)
- origen_id: string(255)
- origen_nivel: int(11)

Habilidades5

- id: PK, int(11)
- nombre: string(200)
- descripcion: string(9999)
- autor: string(200)
- validado: tinyint(1)
- origen_id: string(255)
- origen_nivel: int(11)
- nombre2: string(200)
- descripcion2: string(9999)
- autor2: string(200)
- validado2: tinyint(1)
- origen_id2: string(255)
- origen_nivel2: int(11)
- nombre3: string(200)
- descripcion3: string(9999)
- autor3: string(200)
- validado3: tinyint(1)
- origen_id3: string(255)
- origen_nivel3: int(11)
- nombre4: string(200)
- descripcion4: string(9999)
- autor4: string(200)
- validado4: tinyint(1)
- origen_id4: string(255)
- origen_nivel4: int(11)
- nombre5: string(200)
- descripcion5: string(9999)
- autor5: string(200)
- validado5: tinyint(1)
- origen_id5: string(255)
- origen_nivel5: int(11)

6.2 Login, Control Login y Logout

Para realizar el Login y el Logout hicimos una tabla usuario la cual obtiene un usuario y una contraseña creada específicamente por nosotros. Los usuarios no se pueden loguear y no es posible realizar un registro ya que está controlado.

Formulario de Login visto por pantalla:



Controlador y Template del Login:

Y también hicimos el control Login el cuál lo que se encarga es que no puedas poner por URL ni el Register ni ninguna Ruta protegida por rol de Admin:

```
<?php  
  
//src/controller/LoginController  
  
namespace App\Controller;  
  
  
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;  
use Symfony\Component\HttpFoundation\Response;  
use Symfony\Component\Routing\Annotation\Route;
```

```
use
Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class LoginController extends AbstractController
{
    #[Route('/login', name: 'app_login')]

    public function index(AuthenticationUtils $authenticationUtils): Response
    {
        // get the login error if there is one
        $error = $authenticationUtils->getLastAuthenticationError();

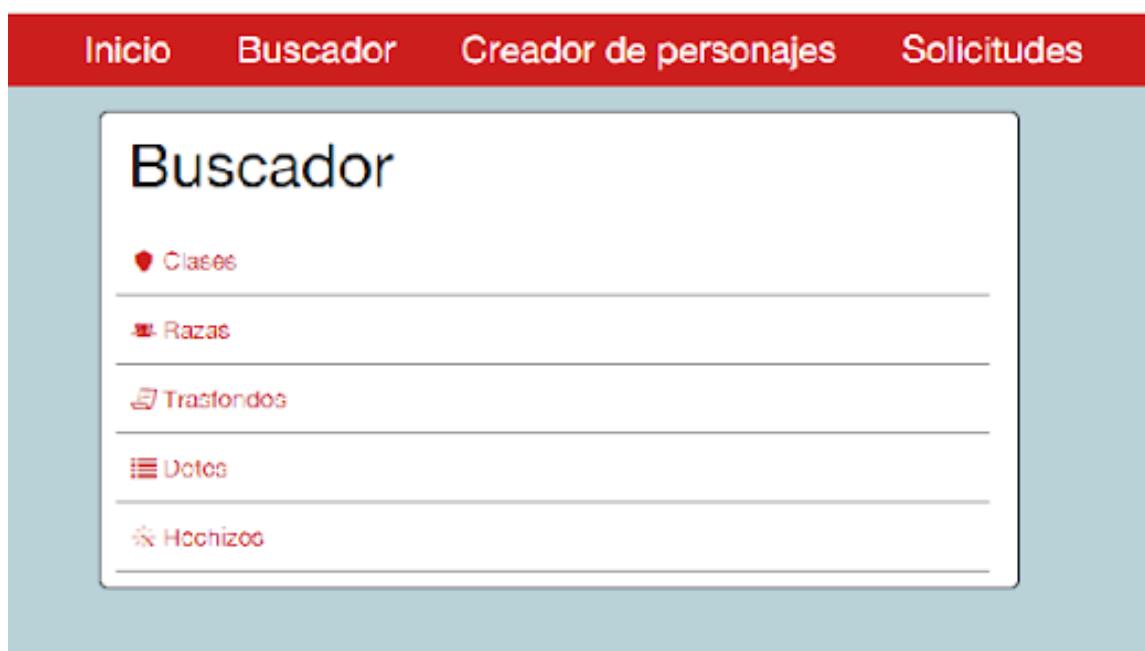
        // last username entered by the user
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('login/index.html.twig', [
            'last_username' => $lastUsername,
            'error'           => $error,
        ]);
    }
}
```

6.3 Creación del Buscador con sus filtros, resultados y pantalla de detalles

En la parte del Buscador, hemos tenido que realizar una implementación de filtros de manera muy exhaustiva. Posteriormente tuvimos que poner una lista con los resultados obtenidos los cuales te muestra sólo la información importante y el propio elemento de la lista es un enlace para llegar a la pantalla final de detalles la cual te muestra todo lo que necesitas saber, desde la descripción hasta tablas de progresión.

Pantalla de Filtros en la Web:



Controlador y Template del Buscador con filtros:

```
/*FORMULARIO DE CLASES*/  
  
#[Route('/buscador_clase', name: 'app_buscador_clase')]  
  
public function Buscador_Clases( Request $request):Response  
{  
  
    $clase=new Clases();  
  
    $form = $this->createForm(ClasesType::Class,$clase);  
  
    $form->handleRequest($request);
```

```
if ($form->isSubmitted() && $form->isValid()) {  
  
    $clases = $form->getData();  
  
    $sesion = $request->getSession();  
  
    $sesion->set('clases', $clases);  
  
    $this->entityManager->flush();  
  
    return $this->redirectToRoute('app_resultados_clases');  
  
}  
  
  
return $this->render(  
'buscador/formularios/clases.html.twig', array('form' => $form));  
}
```

```
{% extends 'base.html.twig' %}

{%
block body %}

<div class="formulario center-h center-v">
<div class="formulario2 ">
    <h1 class="titulo">Dotes</h1>
    {{ form_start(form) }}
    <div class="row">
        <div class="C">
            {{ form_row(form.Nombre) }}
        </div>
        <span class="E"></span>
        <div class="A">
            {{ form_row(form.Requisitos) }}
        </div>
    </div>
    <div class="row">
        <div class="C">
            {{ form_row(form.Beneficios) }}
        </div>
        <span class="E"></span>
        <div class="A">
            {{ form_row(form.Autor) }}
        </div>
    </div>
</div>
```

```
</div>

<div class="row">
    <div class="C">
        <div class="B">
            <div class="A">
                {{ form_row(form.Enviar) }}
            </div>
        </div>
    </div>
    {{ form_end(form) }}</div>
</div>

{%
    % endblock %
}
```

Pantalla de los resultados:

Inicio Buscador Creador de personajes Solicitudes			
	Guerrero <i>Clash of Fates</i>	Salvaciones: Fuerza y Constitución <i>Armadura: Todas y Escudos Armas Simples y 3 Complejas. Elección: 4 Complejas o A Distancia adicionales ó 1 Con Truco Herramientas: Reparación</i>	
	Mago <i>Clash of Fates</i>	Salvaciones: Inteligencia y Sabiduría <i>Armadura: Ninguna Armas: Bastón, Daga, Ballesta Ligera, Dardos y Honda. Herramientas: Ninguna</i>	
	Artista <i>Clash of Fates</i>	Salvaciones: Destreza y Carisma <i>Armadura: Armadura Ligera, Rodela Armas: Simples y 1 Compleja. Herramientas: Ninguna</i>	
	Cultista <i>Clash of Fates</i>	Salvaciones: Sabiduría y Carisma <i>Armadura: Armadura Ligera, Armadura Media, Rodelas, Escudos, Escudo de Torre. Armas: Simples y 2 Complejas. Herramientas: Ninguna.</i>	
	Explorador <i>Clash of Fates</i>	Salvaciones: Fuerza y Destreza <i>Armadura: Armadura Ligera, Armadura Media, Rodela, Escudo Armas Simples, Arco Corto, Arco Largo, Ballesta Ligera, Ballesta de Mano y 1 Compleja. Elección: 1 Compleja adicional ó Ballesta l'esaude. Herramientas: Cazador, Curtidor</i>	
	Picaro <i>Clash of Fates</i>	Salvaciones: Destreza e Inteligencia <i>Armadura: Armadura Ligera Armas Simples, Ballesta de Mano, Katana, Arco Corto, Estoque y Espada Corta. 1 a elegir entre Arco Largo y un arma compleja. Herramientas: Herramientas de Ladrón.</i>	

Plantilla de los resultados:

```
{% extends 'base.html.twig' %}

{% block body %}

    <div class="listado">

        {% for clase in resultado %}

            <a class="elemento" href="{{ path( "app_clase", { "id": clase.id} ) }}>

                {{ clase.Nombre }}</h2>

                <span class="E"></span>

                <i class="C">{{clase.Competencias}}</i>

                <i class="A">{{clase.Autor}}</i>

                <span class="S"><b>Salvaciones:</b>
{{clase.Salvaciones}}</span>

            </a>

        {% endfor %}

    </div>

    {% endblock %}
```

Pantalla de los detalles:

6.4 Creación de Solicitudes con sus filtros adecuados

En esta parte del proyecto, hemos creado un formulario en el cuál dependiendo de la sección es de una manera o de otra y la intención es que los usuarios puedan mandar solicitudes de ideas que tengan para aumentar nuestra enciclopedia, lo cual será revisado por un administrador para darle el visto bueno o no. En esta parte además de meter la solicitud tiene asociado otro formulario el cuál es meter 5 habilidades por idea y está relacionado mediante la entidad de Habilidades5

Pantalla de inicio de Solicitudes y pantalla de uno de los filtros:

The screenshot shows a web application interface. At the top is a red navigation bar with four items: 'Inicio' (Home), 'Buscador' (Search), 'Creador de personajes' (Character Creator), and 'Solicitudes' (Requests). Below the navigation bar is a light blue sidebar titled 'Accesos Rápidos' (Quick Access). The sidebar contains five filter options, each with an icon and a label: 'Clases' (Classes) with a shield icon, 'Razas' (Races) with a dragon icon, 'Trasfondos' (Backgrounds) with a castle icon, 'Dotes' (Abilities) with a grid icon, and 'Hachizos' (Sigils) with a sun icon. Each filter option has a horizontal line below it.

Clases

Nombre

Salvaciones

Requisitos

Magia

No tiene hechizos

Competencias

Puntos de golpe

1d4

Equipamiento

Habilidades a elegir

Img

Autor

[Enviar](#)

[Ir a las solicitudes de las Subclases](#)

Controlador y Plantilla de las Solicitudes:

```
# [Route('/solicitud_clase', name: 'app_solicitud_clase')]

    public function Buscador_Clases( Request $request):Response
    {

        $ipAddress = $request->getClientIp();

        $currentDate = new \DateTime();

        $count =
$this->entityManager->getRepository(Clases::class)->count([
            'ipAddress' => $ipAddress,
            'createdAt' => $currentDate
        ]);

        $clase=new Clases();

        $form = $this->createForm(ClasesType2::Class,$clase);

        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {

            if ($count >= 2) {

                $this->addFlash('error', 'Has superado el limite de
solicitudes por IP');

                return $this->redirectToRoute('app_index');

            }

            $clases = $form->getData();

            if ($clases->getAutor() == null ||
strtolower($clases->getAutor()) == 'clash of fates'){


```

```
    $clases->setAutor('Anónimo');

}

if ($clases->getImg() == null) {

$clases->setImg('https://wallpaperaccess.com/full/1902223.png');

}

$clases->setIpAddress($ipAddress);

$clases->setValidado(false);

$sesion = $request->getSession();

$sesion->set('clases', $clases);

// $this->entityManager->persist($clases);

$this->entityManager->flush();

return

$this->redirectToRoute('app_solicitud_clase_habilidades');

}

return $this->render( 'solicitudes/clases.html.twig',
array('form' => $form));

}
```

```

{%
    extends 'base.html.twig'
%}

{%
    block title %}Hello BuscadorController!{%
    endblock %}

{%
    block body %}

<div class="formulario center-h center-v">
<div class="formulario2 ">
    <h1 class="titulo">Clases</h1>
    {{ form_start(form) }}
    <div class="row">
        <div class="C">
            {{ form_row(form.Nombre) }}
        </div>
        <span class="E"></span>
        <div class="A">
            {{ form_row(form.Salvaciones) }}
        </div>
    </div>
    <div class="row">
        <div class="C">
            {{ form_row(form.Requisitos) }}
        </div>
        <span class="E"></span>
        <div class="A">
            {{ form_row(form.Magia) }}
        </div>
    </div>
</div>

```

```
</div>

</div>

<div class="row">

    <div class="C">
        {{ form_row(form.Competencias) }}
    </div>

    <span class="E"></span>

    <div class="A">
        {{ form_row(form.PuntosDeGolpe) }}
    </div>

</div>

<div class="row">

    <div class="C">
        {{ form_row(form.Equipamiento) }}
    </div>

    <span class="E"></span>

    <div class="A">
        {{ form_row(form.HabilidadesAElegir) }}
    </div>

</div>

<div class="row">

    <div class="C">
        {{ form_row(form.img) }}
    </div>

    <span class="E"></span>

    <div class="A">
```

```
    {{ form_row(form.Autor) }}  
  
  </div>  
  
</div>  
  
<div class="row">  
  
  <div class="C">  
  
    </div>  
  
    <span class="E"></span>  
  
    <div class="A"><br>  
  
      {{ form_row(form.Enviar) }}  
  
    </div>  
  
  </div>  
  
&lt;div&gt;  
  {{ form_end(form) }}  
  
<div class="flex">  
  
  <p><a href="{{ path("app_solicitudes_subclase") }}">Ir a  
las solicitudes de las Subclases</a></p>  
  
  </div>  
  
</div>  
  
</div>  
  
{% endblock %}
```

6.5 Creación de Revisiones con sus resultados

En esta parte del proyecto se podrá ver mediante una query la cual te muestra todo lo que tenga el campo validado a false en secciones y está contabilizado mediante un count() para que sea más accesible para el administrador. Cuando el administrador se mete a cualquiera de las secciones con elementos ya metidos le saldrá una lista con resultados y con botones de Bootstrap los cuáles hará que vea más detalles si así lo desea, le pulse a eliminar solicitud o a validarla.

Pantalla de inicio y de resultados de las Revisiones

The screenshot shows a web application interface. At the top is a red navigation bar with four items: 'Inicio', 'Buscador', 'Creador de personajes', and 'Revisiones'. Below the navigation bar is a light blue sidebar on the left. The sidebar has a title 'Accesos Rápidos' and a list of items with icons and counts:

- 🛡 Clases (2)
- ⌚ Subclases (1)
- 🧟 Razas (0)
- 👤 Subrazas (0)
- ✍ Trastornos (0)
- 📅 Dotes (0)
- ✳ Hechizos (0)

Revisiones de las Clases

	Hola	Ver más	Validar clase	Eliminar clase
	Mercenario	Ver más	Validar clase	Eliminar clase

Controlador de las Revisiones

```
# [Route('/revisiones_clase', name: 'app_revisiones_clase')]

    public function revisiones_clase(): Response

    {

        $revisiones =

$this->entityManager->getRepository(Classes::class)->FindNonValidate
d();

        return

$this->render('revisiones/revisiones_clase.html.twig', [
            'revisiones' => $revisiones,
        ]);
    }
}
```

6.6 Creación de Ficha de Personaje con sus filtros adecuados

En esta sección del proyecto hemos decidido a partir de un formulario con filtros crear una ficha de personaje al usuario la cual va a poder utilizar en sus futuras partidas. Tendrá que imprimirla con **CTRL + P**, para luego darle a descargar como PDF y se le descargará perfectamente bien y lo tenemos bien señalizado el cómo hacerlo en el segundo formulario.

Primer y Segundo formulario y pantalla final visto en la web:

The screenshot shows a web-based character creation interface. At the top, there is a red navigation bar with four items: 'Inicio', 'Buscador', 'Creador de personajes' (which is highlighted in blue), and 'Revisores'. Below the navigation bar is a white form box with a title 'Creador de personajes'. The form contains six input fields arranged in two rows of three. The first row has 'Nombre' (Javier) and 'Nivel' (10). The second row has 'Raza' (Humano) and 'Clase' (Artista). The third row has 'Origen' (Sylfaria) and 'Ocupacion' (Viajero). At the bottom right of the form is a red 'Enviar' button.

Creador de personajes

Subclase

Corriente de la Vida

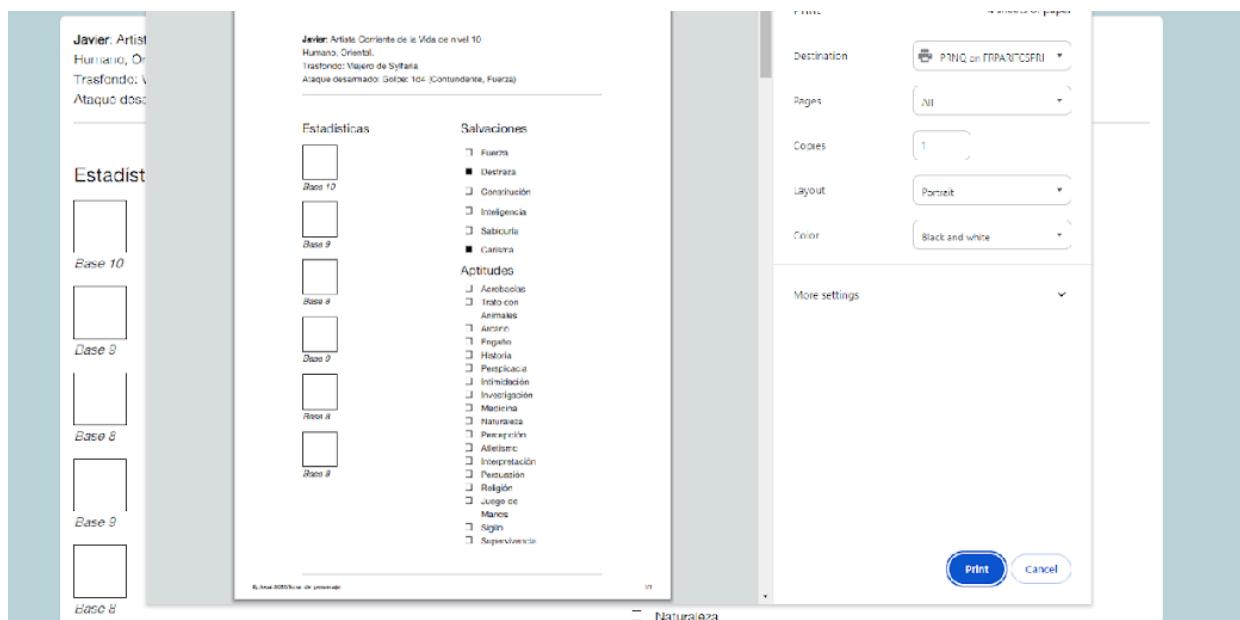
Subraza

Oriental

Tras pulsar enviar se cargará su ficha de personaje.

¡Recuerda usar ctrl + p para guardarla!

Enviar



6.7 Diseño

Para maquetar la aplicación y dejarla bonita hemos decidido principalmente utilizar CSS, pero también hemos usado un poco de Bootstrap.

CSS

```
@import url('../icofont/icofont.min.css');

@font-face {

    font-family: fuente1;

    src: url('../../assets/Fuentes/AlteHaasGroteskRegular.ttf');
```

```
    }

    * {
        margin: 0px;
        padding: 0px;
        text-decoration:none !important;
        box-sizing: border-box;
        font-family: fuente1;
    }
}
```

Bootstrap

```
<button onclick="location.href='{{ path("app_comprobar_clase", {'id': revision.id}) }}'" class="btn btn-info">Ver más</button>

<button onclick="location.href='{{ path('app_validar_clase', {'id': revision.id}) }}'" class="btn btn-success">Validar
clase</button>

<button onclick="location.href='{{ path('app_eliminar_clase', {'id': revision.id}) }}'" class="btn btn-danger">Eliminar
clase</button>
```

6.8 Despliegue de la Aplicación

Finalmente, hemos decidido desplegar nuestra aplicación con el hosting de IONOS, valiéndonos de la herramienta Deploy Now proporcionada por dicho hosting.

7 Presupuesto

Si suponemos un sueldo de 11€/hora y de manera aproximada porque no hemos calculado de manera exacta las horas totales consumidas en el TFG, la inversión total de desarrolladores por tareas sería la siguiente:

Tarea	Horas	Precio
Creación de nombre de la aplicación	0.5	5.5€
Diseño de la Base de Datos	5	55€
Creación del Index	2	22€
Diseño del Login, Logout y Control Login	1	11€
Creación del Buscador	35	385€
Creación de las Solicitudes	20	220€
Inserción de datos en la Base de Datos	50	550€
Revisiones	10	110€
Ficha de Personaje	45	495€
Maquetación y Diseño	34	374€
Despliegue	10	110€
Documentación	25	275€
TOTAL	237.5	2612.5€

8 Conclusiones y vías futuras

Como conclusión, estamos satisfechos con el proyecto que hemos realizado, aunque somos conscientes del margen de mejora y funcionalidad que podríamos haber aprovechado. Entre estas mejoras: controlar mediante registro de IP el número de solicitudes máximas que el usuario puede enviar, (*se intentó implementar pero no fue posible*), además, planteamos como opción

valernos de Tailwind para CSS, pero fue descartado debido a mayor familiaridad con Bootstrap, por último, pensamos que habría sido conveniente que hubiera un botón para descargar automáticamente el PDF de la ficha de personaje una vez generada, pero, fue descartado por dos motivos, pensamos que el usuario debía ser quién tomase esa decisión y los recursos que encontramos para implementarlo según la metodología de Symfony no fueron de nuestro agrado.

Destacar que algunas tablas en la base de datos permanecen vacías pues nos hubiera gustado incluir mayores opciones adicionales a las búsquedas: como Pactos con distintas deidades o que por Clase hubiera una lista de hechizos predefinida, pero debido a la falta de tiempo y focalizar la funcionalidad general de la aplicación tuvimos que descartarla, aunque teníamos un plan para implementarla.

En resumen. Estamos muy contentos con el trabajo realizado, aunque podría haber sido un producto más completo y, debido a nuestro interés personal para con el producto, en el futuro continuaremos mejorando la aplicación.

9 Bibliografía/Webgrafía.

Symfony Documentation. (2005 - Actualidad). *Symfony Docs*. Symfony SAS.

Recuperado de: <https://symfony.com/doc/current/index.html>

Bootstrap. (2010 - Actualidad). *Get started with Bootstrap*. Bootstrap.

Recuperado de: <https://getbootstrap.com/>

Icofont. (2015 - Actualidad). *2400+ free, meticulously crafted icons*.

Recuperado de: <https://icofont.com/>

iStock. (). *Modifica imágenes de iStock con IA con unos pocos clics!*

Recuperado de: <https://istockphoto.com>

Nivel20. (2024). *Nivel20*. Recuperado de: <https://nivel20.com/>