

PROYECTO Final

DnDMaker



Adrià García Pons
IES Punta del Verde
Administración de Sistemas Informáticos en Red
adriagarciapons3@gmail.com
06/06/2022



Índice

1. Introducción

- a. ¿Por qué DnDMaker?
- b. Explicación General del Proyecto

2. Desarrollo del proyecto

- a. Como se ha organizado el trabajo
- b. Índice de herramientas utilizadas

3. Backend

- a. Infraestructura del proyecto
- b. Introducción al esquema de datos del Proyecto
- c. Rutas empleadas en el proyecto
- d. Sistema de caché y código reseñable

4. Frontend

- a. Visualizado de las distintas páginas del proyecto

5. Áreas de mayor innovación

- a. Auth0
- b. AWS

6. Conclusión

- a. ¿Qué queda pendiente?

7. Webgrafía



1. Introducción

¿Por qué DnD Maker?

Realizar un proyecto de esta magnitud no debiera verse como obligación sino como oportunidad, DnD Maker no es un proyecto, es la **solución a un problema real**, que, además, me ha permitido desarrollarme en tecnologías que durante el curso no pude tanto como me hubiese gustado.

El propósito de DnD Maker, como he dicho, es el de **cubrir una necesidad existente**, y una vez se avance aún más en su desarrollo barajo la posibilidad de subirlo de forma pública bajo un dominio.

Sin embargo, ¿Qué es DnD Maker?, ¿Qué significan sus siglas?.

DnD Maker significa literalmente traducido “Dragones y Mazmorras Creador”, y es que Dragones y Mazmorras es una aglomeración de distintos tipos de juegos de rol...

Los juegos de rol se remontan a hace siglos, pero en el formato en el que los conocemos surgieron de algo tan humano como desempeñar un rol o papel, jugar a algo en cooperativo con más personas y en un formato parecido al de un juego de mesa.





Con la llegada de la informática e internet los juegos de rol ya no son solo jugados de forma local, como si de un juego de mesa se tratara.

Actualmente decenas de miles de personas juegan a través de Internet (entre ellos yo).



Sin embargo, la llegada del internet ha fomentado la marcha del cuaderno de papel, en el que teníamos toda la información de nuestra partida centralizada... Ahora nuestra información se encuentra repartida en distintos chats, los sitios webs en los que desarrollamos las partidas, documentos abandonados en algún directorio de nuestro dispositivo, etc.

La idea del proyecto es permitir a cualquiera que la utilice centralizar toda esa información.



Explicación General del proyecto

El proyecto se divide en dos apartados: El apartado Software (el desarrollo en angular) y el montaje de una infraestructura productiva en AWS.

Para la parte de angular cabe a resaltar que la temática como se ha dicho con anterioridad gira en torno a dragones y mazmorras mientras que para el apartado de la infraestructura veremos como hemos trabajando en el mantenimiento de servicios como nginx, implementación de DNS, servidores de balance de carga, etc.

Ambas partes juegan un papel fundamental en el proyecto y son necesarias para entender lo que se pretendía realizar (lo cual, finalmente, se ha realizado).

Aún queda mucho pendiente de hacer, sobretodo en el apartado de angular, pero el proyecto lo considero sólido.

Lo principal que he buscado con este proyecto ha sido innovar y aprovechar la oportunidad que se me planteaba para aprender todo aquello por lo que sentía curiosidad, montar instancias y servidores en AWS... Profundizar en el manejo de usuarios de Angular, etc.



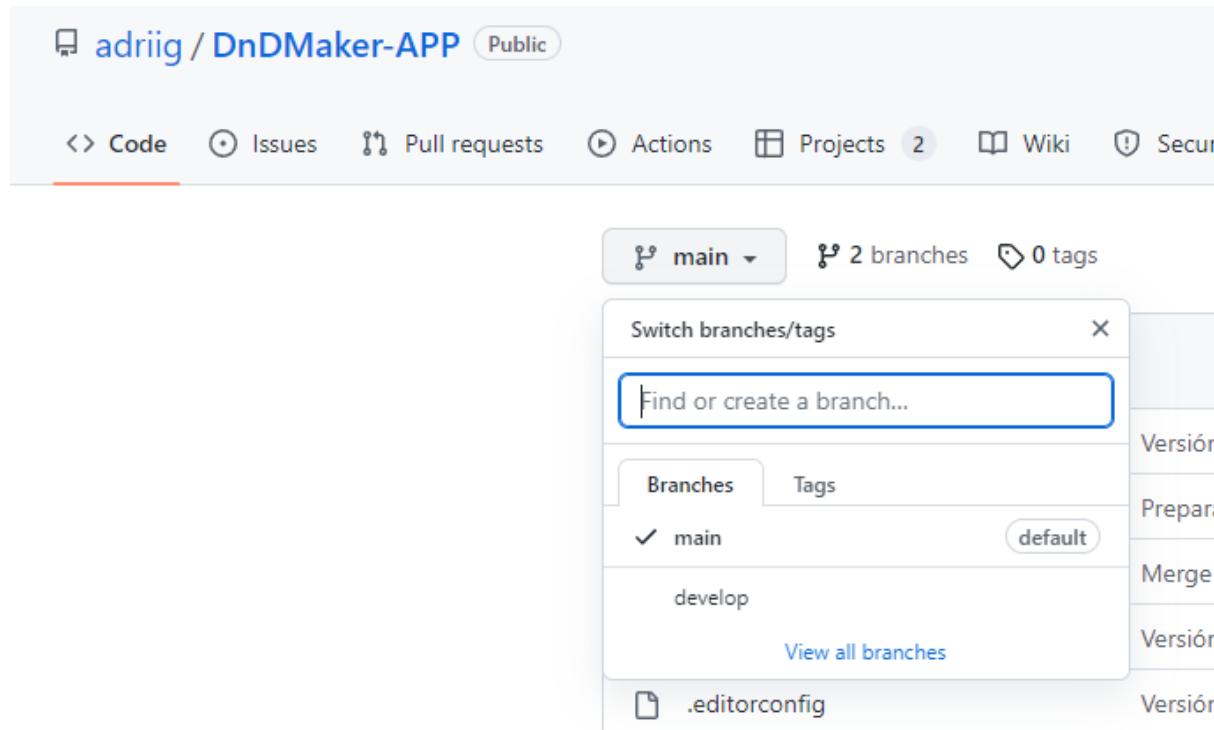


2. Desarrollo del Proyecto

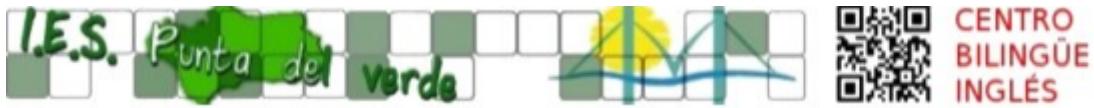
¿Cómo se ha organizado el proyecto?

He decidido desarrollar el proyecto imitando el como funcionan las empresas y equipos de desarrollo profesionales.

He creado 2 branches para el repositorio de APP (Angular), una de dev (en la que subo versiones en las que no hay funcionalidades completas) y una Main (en la que se sube el progreso bien actualizado).



The screenshot shows a GitHub repository page for 'adriig / DnDMaker-APP'. The 'Code' tab is selected. At the top right, there are buttons for 'main', '2 branches', and '0 tags'. A dropdown menu titled 'Switch branches/tags' is open, showing a search bar with 'Find or create a branch...' and a list of branches. The 'main' branch is checked and labeled 'default'. The 'develop' branch is also listed. Below the branches, there is a link to 'View all branches'. To the right of the branches, there are several small cards with text like 'Versiór', 'Prepar:', 'Merge', 'Versiór', and 'Versiór'.



Tambien he creado una ToDoList en github para organizar el trabajo restante

The screenshot shows a GitHub project page for 'adrig / DnDMaker-APP' with a public status. The main navigation bar includes Code, Issues, Pull requests, Actions, Projects (selected), Wiki, Security, Insights, and Settings. Below the navigation is a 'ToDoList' section with the following columns:

- GENERAL** (5 cards):
 - Acabar la sección de Games
 - Creación de habilidades para Razas y Clases**
 - Agregar ventana emergente con angular sobre solicitud de join a un game.
 - Subir a Heroku // Hacer Deploy
 - Opción de editar en characters:
 - Agregar botón en el check del character
 - Que te guarde el valor de tus variables
- To Do CSS** (10 cards):
 - Agregar en la navbar un ngif para cada opción del menu en la que según en qué componente estés este con un color distinto
 - Arreglar que el fondo de pantalla se desplace en lugar de ser una imagen fija
 - Agregar paginación a las tablas
 - Agregar títulos a cada html de los componentes
 - Crear los enlaces correspondientes a los distintos componentes
 - Implementar todos los fondos de pantalla a cada página
 - Mejorar CSS para la creación de objetos ***
 - ...
 - ...
 - ...
- In progress** (1 card):
 - Implementar que se vea nombre de usuario*** en el listado de Character. Classes. Races... (En lugar de id)
- STOPPED** (2 cards):
 - Aregar la creación de Characters:
 - Selección de clase
 - Selección de raza
 - Habilidades/Stats
 - Agregar Resources:
 - CRUD Hechizos
 - CRUD Razas
 - CRUD Clases
- Done** (1 card):
 - Crear base de datos de usuarios que funcione con auth0:
 - _idUser
 - _username

Al hacer el login te genere un usuario aleatorio para luego poder editarla desde el profiler

Además, en ese sistema hemos ido empleando un sistema de acotación del tipo 1.X.Y

Siendo Y las distintas versiones que añaden/corren contenido para las versiones X que son las que aportan funcionalidades completas.

The screenshot shows the commit history for the 'main' branch of the 'DnDMaker-APP' repository. The most recent commit is '13.6' by 'adrig' (4 hours ago), which merges changes from the 'main' branch of 'https://github....'. The commit message is 'Mejoras varias del CSS Opción de subir imágenes al crear personajes'. The code diff shows modifications in several files, including 'src/app/components/check.component.ts', 'src/app/components/create.component.html', 'src/app/components/create.components.ts', 'src/app/components/forum.component.css', 'src/app/components/forum.component.html', 'src/app/models/character/characters.ts', and 'src/app/services/characters.service.ts'. The commit also includes a note: 'Merge branch 'main' of https://github....'.



Herramientas utilizadas

1. Servidores:

- a. **AWS** → Infraestructura del proyecto (No estudiado antes)
- b. **Vercel.com** → Aquí se ha hecho deploy del proyecto (Similar a aplicaciones dadas en bases de datos).

2. Servicios S.O:

- a. **nginx** → Para la implantación de la página web en la instancia de AWS (Estudiado en SRI)
- b. **openssl** → Manejo de certificados para lograr montar las páginas en el servicio https (Estudiado en SRI)
- c. **npm** → Gestor de dependencias de Node (Lo hemos estudiado en Bases de datos)
- d. **tsc** → Compilador para typescript (Estudiado en base de datos)
- e. **git** → Gestor de repositorios (Estudiado en bases de datos)

3. Aplicaciones:

- a. **putty** → Conexión por SSH con máquinas virtuales (No estudiado en profundidad).
- b. **angular** → Empleado para el montaje general del proyecto (Estudiado en bases de datos).
- c. **postman** → Utilizado para pruebas, atacando a la API-Rest (Estudiado en base de datos).
- d. **github** → Para la gestión de los repositorios de forma mucho más visual que git (No estudiado antes).
- e. **Filezilla** → Envío de archivos a la instancia por SFTP (No estudiado antes).

4. Aplicaciones Angular:

- a. **Angular Material** → Para el diseño general de algunos componentes del proyecto (Estudiado en bases de datos).
- b. **auth0** → Gestión de sesiones para usuarios (No estudiado antes)
- c. **uuidv4** → Generación de id's para objetos (No estudiado antes)
- d. **pagination** → Paginación para tablas o listas (No estudiado antes)

5. Componentes Framework:

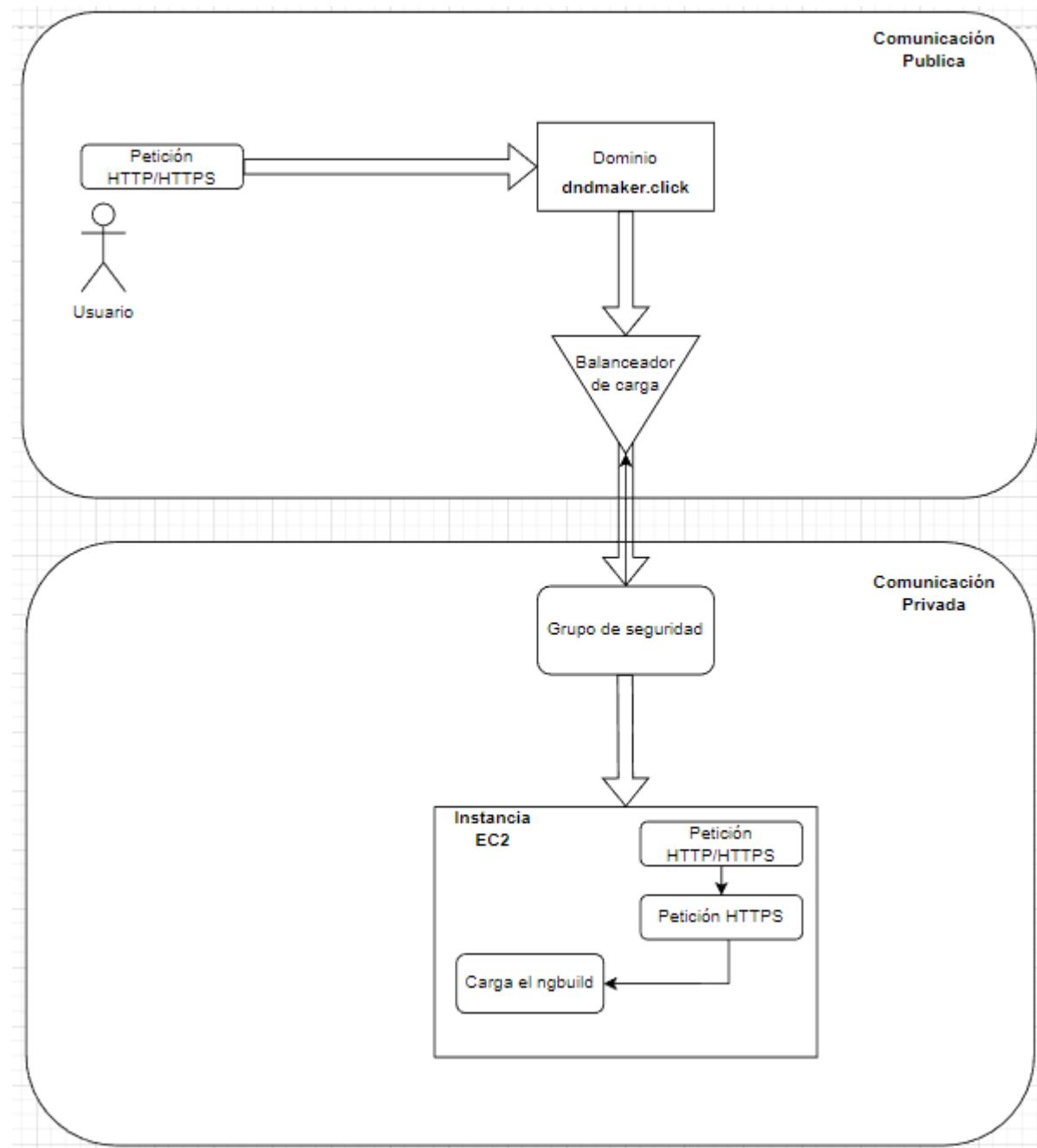
- a. **Instancias AWS EC2** → Instancia Virtual en la que se ha desplegado el servidor que hospeda al proyecto (No estudiado antes)
- b. **Balanceador de carga** → Responsable de gestionar el flujo de peticiones entrantes entre las máquinas objetivo (No estudiado antes)
- c. **Grupos de Seguridad** → Espacios para configuración de ciberseguridad para los miembros del grupo tales como los puertos/servicios permitidos o las ip's permitidas (No estudiado antes).
- d. **Route53 & Gestor Certificados** → Compra y configuración de un dominio propio e implementación del mismo en el proyecto (No estudiado antes)



3. Backend

Infraestructura del proyecto

Este sería un esquema simplificado de cómo funciona la infraestructura en AWS:





El usuario hará su petición, sin importar que sea http o https, la hará normalmente al dominio del proyecto “dndmaker.click”, el cual, está configurado para redireccionar lo que le llegue al balanceador.

El balanceador de carga del proyecto ya revisa que es la petición que le llega y solo aceptará o bien http o https. (En el balanceador es donde he tenido que configurar los certificados de dominio). El balanceador derivará esta petición al “target group”, que, aunque en mi caso solo se forme por una instancia podría estar perfectamente integrado por más de una.

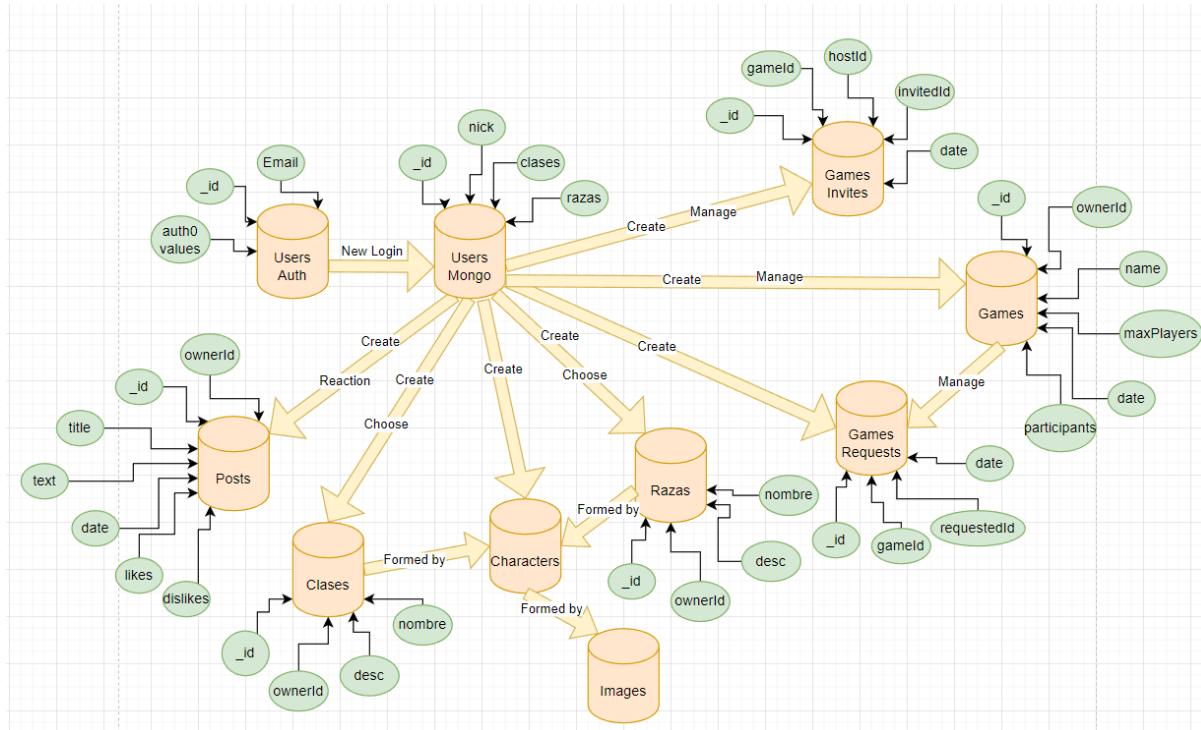
La comunicación entre balanceador y la instancia ya se realiza por las IP's privadas / DNS privados.

nginx se encargará de transformar la petición entrante de http a https y, gracias a openssl las páginas del proyecto contarán con certificado autofirmado.

El servicio nginx está configurado para que tome como directorio raíz el que contiene los archivos resultantes del “ng build” que habremos tenido que hacer con anterioridad a nuestro proyecto

Al usuario se le enviará el cuerpo html correspondiente a su petición original.

Introducción al esquema de datos del proyecto



Como podemos observar, prácticamente todas las colecciones de BDD penden de “Users Mongo”, la cual viene a ser la principal.

Cada vez que un usuario se loggea por primera vez generamos una nueva entrada para esta colección, lo hacemos así ya que auth0 nos limita la cantidad de información que podemos guardar en su BDD y, también, la cantidad de campos que pueden albergar los usuarios.

Ahora explicaré parte por parte las relaciones que vemos en pantalla.



POSTS

Esta colección es la encargada del foro, cada entrada es un post en el foro. Como podemos ver, tiene campos para texto, título y por supuesto recoge su propio id para diferenciarse del resto de entradas del foro y el de su creador.

También tiene un campo de “tipos” con el que podremos diferenciarlo del resto de posts.

Los usuarios podrán crear Posts o reaccionar a ellos con like/dislikes, los cuales son arrays que registran cada usuario que interactúe.

CLASES y RACES

Las clases y razas son campos necesarios para nuestro personaje, podremos crearlas nosotros mismos y usarlas en nuestro personaje o podremos importarlas de otros usuarios para emplearlas.

CHARACTER

La clase character se construye a partir de raza, clase y una serie de otros campos. También podremos adjuntar imágenes en ella ya que se comunica con la colección de images.

GAMES

Los games son el apartado más complejo del proyecto, se dividen en varios componentes, desde el visualizado general de los games, tus propios games, o el sistema de magament de un game en específico.

Dentro de los games como administradores de juego podremos invitar a usuarios y gestionar las solicitudes que hagan los usuarios para entrar en el game.

Como usuarios podremos solicitar la entrada a todos los games que no sean nuestros y, también, aceptar/denegar invitaciones de juego.



Rutas empleadas en el proyecto

Dentro de Angular hemos utilizado las siguientes rutas para gestionar los componentes:

```
5 const routes: Routes = [
6   { path: '', component: DashboardComponent, children: [
7     { path: '', redirectTo: "forum", pathMatch: 'full'},
8     { path: '', component: AuthenticationButtonComponent},
9     { path: '', component: LoginButtonComponent},
10    { path: '', component: LogoutButtonComponent},
11    { path: 'forum', component: ForumComponent},
12    { path: 'games', component: GamesComponent},
13    { path: 'games/manage', component: ManagementComponent},
14    { path: 'games/view/:id', component: GameComponent},
15    { path: 'characters', component: CharactersComponent},
16    { path: 'characters/detail/:id', component: CheckComponent},
17    { path: 'characters/create', component: CreateComponent},
18    { path: 'guides', component: GuidesComponent},
19    { path: 'profile', component: ProfileComponent},
20
21    { path: 'resources/classes', component: ClassesComponent},
22    { path: 'resources/classes/create', component: CreateClassComponent},
23    { path: 'resources/races', component: RacesComponent},
24    { path: 'resources/classes/create', component: CreateClassComponent},
25    { path: 'resources/races/create', component: CreateRacesComponent},
26    { path: 'resources/races/detail/:id', component: CheckRacesComponent }
27  ]}
28];
29
```



Dentro de la API-Rest se han utilizado todas las siguientes rutas:

```
misRutas(){
    this._router.get('/Razas/get', this.getRazas)
    this._router.post('/Razas/Sub/add', this.addSubRaza)
    this._router.post('/Razas/add', this.addRaza)
    this._router.get('/Razas/search/:id', this.searchRaza)
    this._router.delete('/Razas/delete/:id', this.deleteRaza)
    this._router.get('/Razas/getmy/:idOwner', this.getmyRaces)
    this._router.get('/Razas/getpickable/:idOwner/:array', this.getpickableClasses)

    this._router.get('/Users/get', this.getUsers)
    this._router.post('/Users/add', this.addUser)
    this._router.delete('/Users/delete', this.deleteUser)
    this._router.get('/Users/search/:id', this.searchUser)

    this._router.get('/Users/getClasses/:id', this.getClassesOfUser)
    this._router.put('/Users/addClass/:id/:valueId', this.addClassToUser)
    this._router.get('/Users/existClassInUsers/:id/:valueId', this.existClassInUser)
    this._router.get('/Users/deleteClassInUser/:id/:valueId', this.removeClassInUser)

    this._router.get('/Users/getRaces/:id', this.getRaceOfUser)
    this._router.put('/Users/addRaces/:id/:valueId', this.addRaceToUser)
    this._router.get('/Users/existRaceInUser/:id/:valueId', this.existRaceInUser)
    this._router.get('/Users/deleteRaceInUser/:id/:valueId', this.removeRaceInUser)

    this._router.get('/Spells/get', this.getSpells)
    this._router.post('/Spells/add', this.addSpells)
    this._router.delete('/Spells/delete', this.deleteSpells)
    this._router.get('/Spells/search/:id', this.searchSpells)

    this._router.get('/Characters/get', this.getCharacters)
    this._router.get('/Characters/getmy/:idOwner', this.getmyCharacters)
    this._router.get('/Characters/search/:id', this.searchCharacter)
    this._router.post('/Characters/add', this.addCharacter)
    this._router.delete('/Characters/delete/:id', this.deleteCharacter)

    this._router.get('/Classes/get', this.getClass)
    this._router.post('/Classes/add', this.addClass)
    this._router.post('/Classes/search/:id', this.searchClass)
    this._router.delete('/Classes/delete/:id', this.deleteClass)
    this._router.get('/Classes/getmy/:idOwner', this.getmyClasses)
    this._router.get('/Classes/getpickable/:idOwner/:array', this.getpickableClasses)
```

```
this._router.get('/Posts/get', this.getPosts)
this._router.get('/Posts/getmy/:idOwner', this.getmyPosts)
this._router.get('/Posts/search/:id', this.searchPost)
this._router.delete('/Posts/delete/:id', this.deletePost)
this._router.post('/Posts/add', this.addPost)
this._router.get('/Posts/getptype/:type', this.getPostPerType)
this._router.put('/Posts/addComment/:id', this.addComment)
this._router.get('/Posts/like/:id/:idvalue', this.likePost)
this._router.get('/Posts/dislike/:id/:idvalue', this.dislikePost)
this._router.get('/Posts/unlike/:id/:idvalue', this.unLikePost)
this._router.get('/Posts/undislike/:id/:idvalue', this.unDislikePost)

this._router.get('/games/get', this.listGames)
this._router.get('/games/get/:gameId', this.listGame)
this._router.get('/games/from/:owner', this.listGamesFromOwner)
this._router.post('/games/create', this.createGame)
this._router.delete('/games/delete/:gameId', this.deleteGame)

this._router.put('/games/participant/delete', this.deleteParticipant)
this._router.put('/games/participant/add', this.addParticipant)

this._router.get('/games/request/from/:ownerId', this.getGameRequestsFromOwner)
this._router.get('/games/request/from/:ownerId/:gameId', this.getGameRequestsFromOwnerInGame)
this._router.get('/games/request/get/:gameId', this.getGameRequestsFromGameId)
this._router.post('/games/request/create', this.createGameRequest)
this._router.delete('/games/request/delete/:ownerId/:gameId', this.deleteGameRequest)

this._router.get('/games/invite/get/invited/:invited', this.getInvitesFromInvited)
this._router.get('/games/invite/get/host/:host', this.getInvitesFromHost)
this._router.get('/games/invite/get/game/:gameId', this.getInvitesFromGameId)
this._router.post('/games/invite/create', this.createInvite)
this._router.put('/games/invite/accept', this.acceptInvite)
this._router.delete('/games/invite/delete/:inviteId', this.deleteInvite)
this._router.delete('/games/invite/delete/game/:gameId', this.deleteInviteFromGame)
```



Sistema de caché y código reseñable

1. **Ahorro de código en API-Rest:** A modo de ahorrar código en la API-Rest se ha creado un método el cual nos sirve para ahorrarnos código y simplificar el código:

```
private executeMongoQuery = async (req: Request, res: Response, callback: Function) => {
    await db.conectarBD()
        .then(async () => {
            callback()
        })
        .catch((mensaje) => {
            res.send(mensaje)
        })
}
```

Ejemplo de su uso siendo llamado por otros 2 métodos:

```
private addParticipant = async (req: Request, res: Response) => {
    const { gameId, playerId } = req.body
    await this.executeMongoQuery(req, res, async () => {
        const query = await GameDB.findOneAndUpdate(
            {
                _id: gameId
            },
            {
                $push: {
                    participants: playerId
                }
            }
        );
        res.json(query)
    });
}

private acceptInvite = async (req: Request, res: Response) => {
    const { gameId, invited } = req.body
    await this.executeMongoQuery(req, res, async () => {
        const query = await GameDB.findOneAndUpdate(
            {
                _id: gameId
            },
            {
                $push: {
                    participants: invited
                }
            }
        );
        res.json(query)
    });
}
```

- 2. Sistema de caché:** Se ha utilizado por ejemplo en el foro, en primer lugar, declaramos Maps que serán los retenedores de la información

```
postOwners: Map<string, Users> = new Map<string, Users>();
postReaction: Map<string, Number> = new Map<string, Number>();
```

Aquí, en este extracto vemos que ya hemos realizado la petición para obtener todos los posts y ahora con todos reunidos en una variable la recorremos, instanciamos los posts y mediante un método de la clase “existReaction” obtenemos nuestro “feedback” el cual introduciremos en el caché para luego ser mostrado en el html.

```
this.posts.forEach((data: { _id: string; _Titulo: string; _Texto: string; _Likes: string[]; _Dislikes: string[]; _Date: Date; }) =>
  const post = new Posts(data._id, data._Titulo, data._Texto, data._Likes, data._Dislikes, data._Date, data._Tipo, data._Owner);
  console.log(post);
  let feedback=post.existReaction(this.profileId)
  this.postReaction.set(data._id, feedback);
);
```

Aquí tendríamos el ejemplo del método encargado en resolver este caché

```
resolveReaction(id: string) {
  return this.postReaction.get(id)
}
```

Y su respectiva llamada desde el html:

```
<ng-template #noParticipants>
  <p class="subtitle">No participants found yet</p>
</ng-template>
</div>
```

- 3. Utilización de ng-template:** Como vemos aquí, ng-template nos permite en función del valor de una concretada en el ngif superior hacer el display o no de un determinado extracto de html. Permitiéndonos ahorrar código y tener todo más ordenado y legible.

```
<div class="card">
  <ng-template #noParticipants>
    <p class="subtitle">No participants found yet</p>
  </ng-template>
</div>
<div class="invites">
  <div class="card">
    <h1 class="title">PENDING REQUESTS</h1>
    <table class="games-table" *ngIf="getRequests().length > 0; else noRequests">
      <tr>
        <th colspan="2">PLAYER</th>
      </tr>
      <tr *ngFor="let invite of getRequests()">
        <td>{{ resolveUsername(invite.requester) }}</td>
        <td>
          <mat-icon style="color: #rgb(154, 186, 152);> (click)="acceptRequest(invite.requester)">check_circle</mat-icon>
          <mat-icon></mat-icon>
          <mat-icon style="color: #rgb(197, 91, 91);> (click)="cancelRequest(invite.requester)">cancel</mat-icon>
        </td>
      </tr>
    </table>
    <ng-template #noRequests>
      <p class="subtitle">No requests found yet</p>
    </ng-template>
  </div>
</div>
```



- 4. Paginación:** En todas las tablas que he visto que era necesario he aplicado paginación, como se puede ver en la captura el modo de implementarlo ha sido mediante “pagination-controls” al final de la página en el que se concretaba que hacer en caso de detectar un cambio y, en la parte superior, un “tr” el cual funciona a modo de lista y en él concretamos el número de elementos a mostrar y, mediante id y nombre, lo relacionamos con el pagination-control

```

<tr *ngFor="let Characters of listMyCharacters | paginate: { id: 'pagination2', itemsPerPage: 8, currentPage: Mypages }">
  <td>
    <div class="Contorno">{{ Characters._NombrePersonaje }}</div>
  </td>
  <td>
    <div class="Contorno">{{ Characters._Raza }}</div>
  </td>
  <td>
    <div class="Contorno">{{ Characters._Clase }}</div>
  </td>
  <td>
    <div class="Contorno">{{ resolveUsername(Characters._IdOwner) }}</div>
  </td>
  <td>
    <a (click)="deleteCharacter(Characters._id)" matTooltip="Eliminar Usuario" style="cursor: pointer">
      <mat-icon>delete</mat-icon>
    </a>
    <mat-icon routerLink="detail/{{Characters._id}}">search</mat-icon>
  </td>
</tr>
</table>
<pagination-controls class="pagination" id="pagination2" (pageChange)="Mypages = $event"></pagination-controls>

```

- 5. Utilización de uuidv5:** Esta herramienta me ha permitido generar los id's de forma que no sean nunca coincidentes y, por supuesto, configurables.

Su utilización es fácil, basta con importar su respectivo módulo tras instalar

```

import { UserService } from 'src/app/';
import { v4 as uuidv4 } from 'uuid';

```

Y ya podrá ser utilizado con normalidad

```

const invite = new GameInvite(
  uuidv4(),
  this.gameId,
  this.game.host,
  userId,
  new Date()
)

```

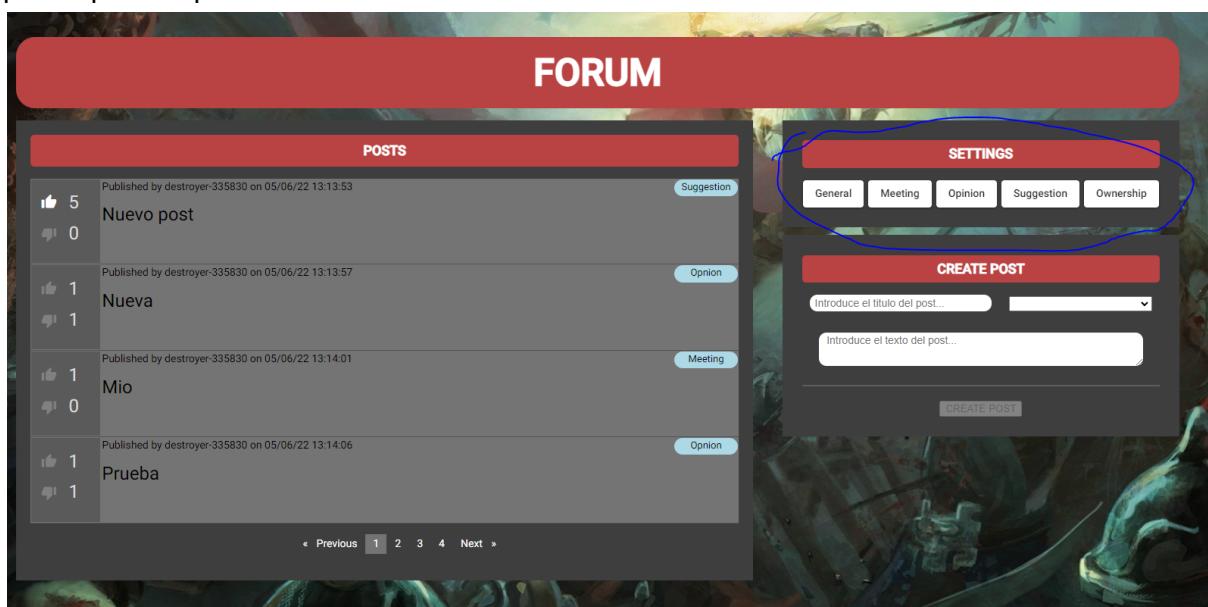


4. Frontend

Visualizado de las páginas del proyecto

1. FORUM

Como vemos en la siguiente captura desde nuestra página podremos diferenciar entre los distintos tipos y también podremos filtrar por “Ownership”, es decir, nos listará solo aquellos posts que nos pertenecen.



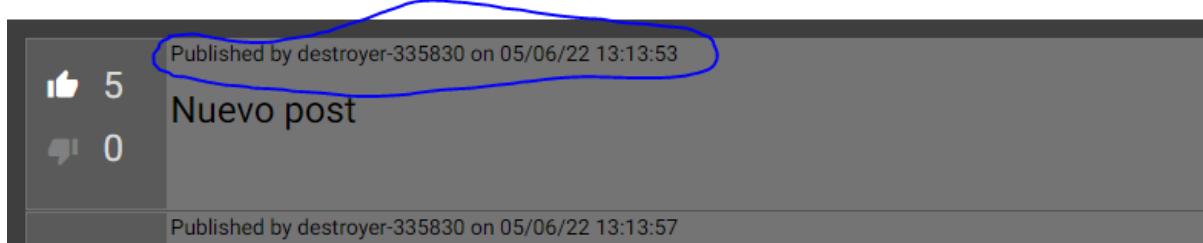
The screenshot shows a forum interface. On the left, there's a list of posts with titles like "Nuevo post", "Nueva", "Mio", and "Prueba". Each post has a like count (e.g., 5, 1, 1, 1) and a dislike count (0, 1, 0, 1). On the right, a modal window titled "SETTINGS" is open, showing tabs for "General", "Meeting", "Opinion", "Suggestion", and "Ownership". A blue oval highlights the "Ownership" tab. Below the settings, a "CREATE POST" form is visible with fields for title and text, and a "CREATE POST" button.

También tenemos opción a interactuar con los posts dejando un like o dislike. La página está diseñada para actualizarse al instante ya que modifica de forma simultánea el html para que a vista de usuario los cambios sean instantáneos mientras que de forma paralela se encarga de hacer una petición a base de datos.

Como vemos para el método del like, hace una comprobación con otro método para analizar el estado (Like/Dislike/Neutro), y si estaba el dislike puesto lo quita, para que no estén ambos simultáneamente, y, en el foreach que ejecuta lo que hace es el array que carga el componente a modo de caché con todo el contenido del foro le realiza un push, ya que los likes/dislikes son realmente un Array y lo que se muestra no es más que un Array.length.

```
like(id: string) {
  if(this.resolveReaction(id)==-1) {
    this.unDislike(id)
  }
  this.postService.like(id, this.profileId).subscribe(data => {
  });
  this.posts.forEach((element: { _id: string, _Likes: Array<string> }) => {
    if(element._id==id) {
      element._Likes.push("")
    }
  });
  this.postReaction.set(id, 1);
}
```

Podemos ver que la fecha de publicación sale con formato, y el nombre del usuario sale en texto en lugar de id que es lo que se guarda en la colección de post.



Lo anteriormente explicado se debe a que resolvemos el nombre de usuario a partir del id del usuario. Al ejecutarse por primera vez el componente cacheamos todos los id's de usuarios y les asignamos su respectivo nombre de usuario en un hashmap.

La fecha le damos un mejor formato instanciando la fecha del mongo como un “New date” y generando así un “ISODate” con el formato que le pasemos como parámetro.

```
<div class="caratula">
  <div class="InfoCreacion">
    Published by {{resolveUsername(post._IdOwner)}} on {{ stringAsDate(post._Date) | date:'dd/MM/yy
    HH:mm:ss' }}
  </div>
  <div class="InfoTipo">
```

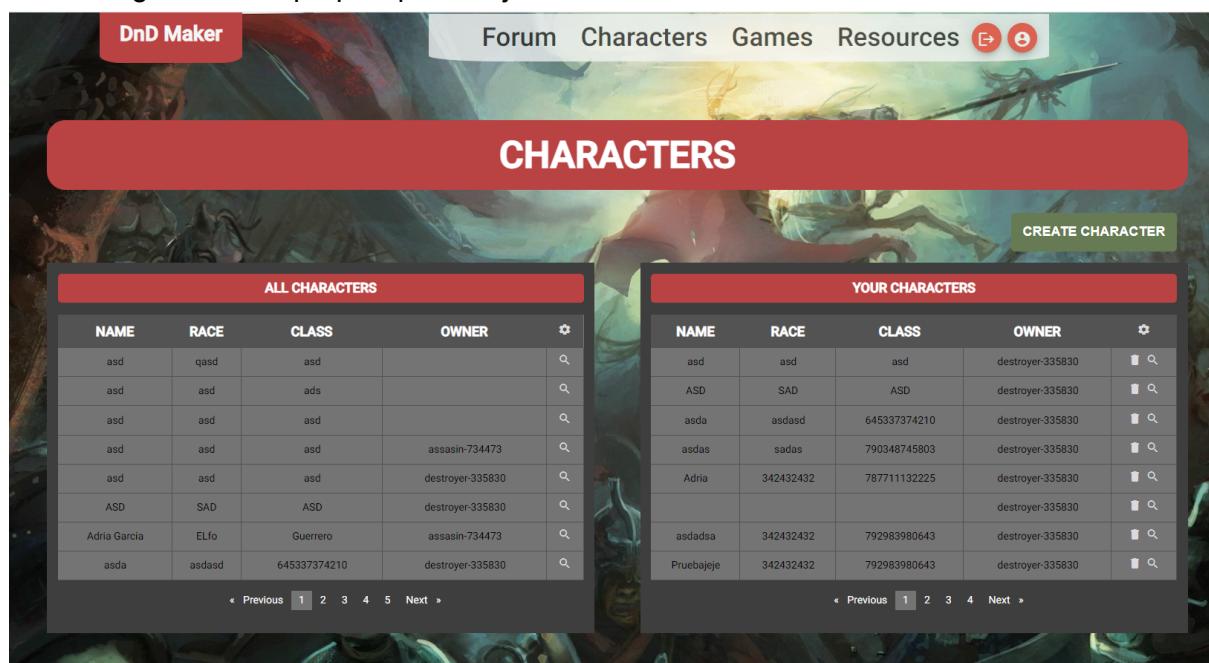


Los posts están ordenados en función del número de likes y esto ocurre gracias a la query que se realiza desde la API-Rest

```
private getPosts = async (req: Request, res: Response) => {
    await db.conectarBD()
    .then( async (mensaje) => {
        console.log(mensaje)
        const query = await PostsDB.aggregate([
            { $project: {"_id":1, "_Titulo":1, "_Date":1, "_Texto":1, "_Likes":1,
                "_Dislikes":1, "_Tipo":1, "_IdOwner":1, "_Comentarios":1,
                LikesNumber: {$size: "$_Likes"}},
            },
            {$sort: {LikesNumber: -1}}
        ])
        res.json(query)
    })
    .catch((mensaje) => {
        res.send(mensaje)
    })
}
```

2. CHARACTERS

Esta es la visión general que tenemos en la pantalla de 'Characters', podemos ver dos tablas, una para todos los personajes sin importar el dueño y, en la derecha, una que tan solo alberga nuestros propios personajes.



| ALL CHARACTERS | | | | |
|----------------|--------|--------------|------------------|--|
| NAME | RACE | CLASS | OWNER | |
| asd | qasd | asd | | |
| asd | asd | ads | | |
| asd | asd | asd | | |
| asd | asd | asd | assassin-734473 | |
| asd | asd | asd | destroyer-335830 | |
| ASD | SAD | ASD | destroyer-335830 | |
| Adria Garcia | ELfo | Guerrero | assassin-734473 | |
| asda | asdasd | 645337374210 | destroyer-335830 | |

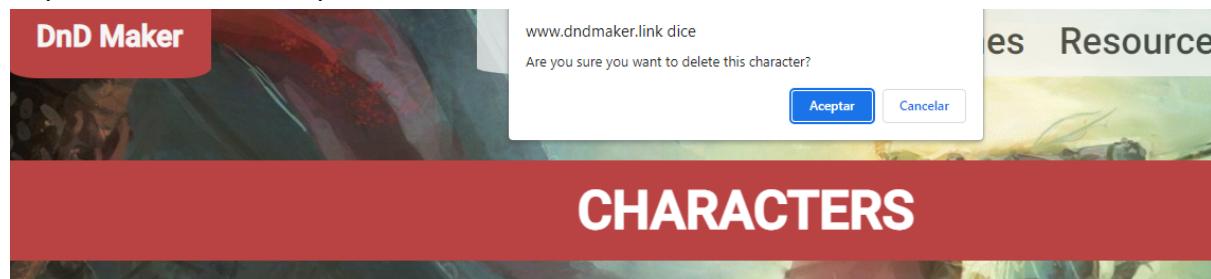
< Previous 1 2 3 4 5 Next >

| YOUR CHARACTERS | | | | |
|-----------------|-----------|--------------|------------------|--|
| NAME | RACE | CLASS | OWNER | |
| asd | asd | asd | destroyer-335830 | |
| ASD | SAD | ASD | destroyer-335830 | |
| asda | asdasd | 645337374210 | destroyer-335830 | |
| asdadas | 342432432 | 790348745803 | destroyer-335830 | |
| Adria | 342432432 | 787711132225 | destroyer-335830 | |
| | | | destroyer-335830 | |
| asdadasa | 342432432 | 792983980643 | destroyer-335830 | |
| Pruebajeje | 342432432 | 792983980643 | destroyer-335830 | |

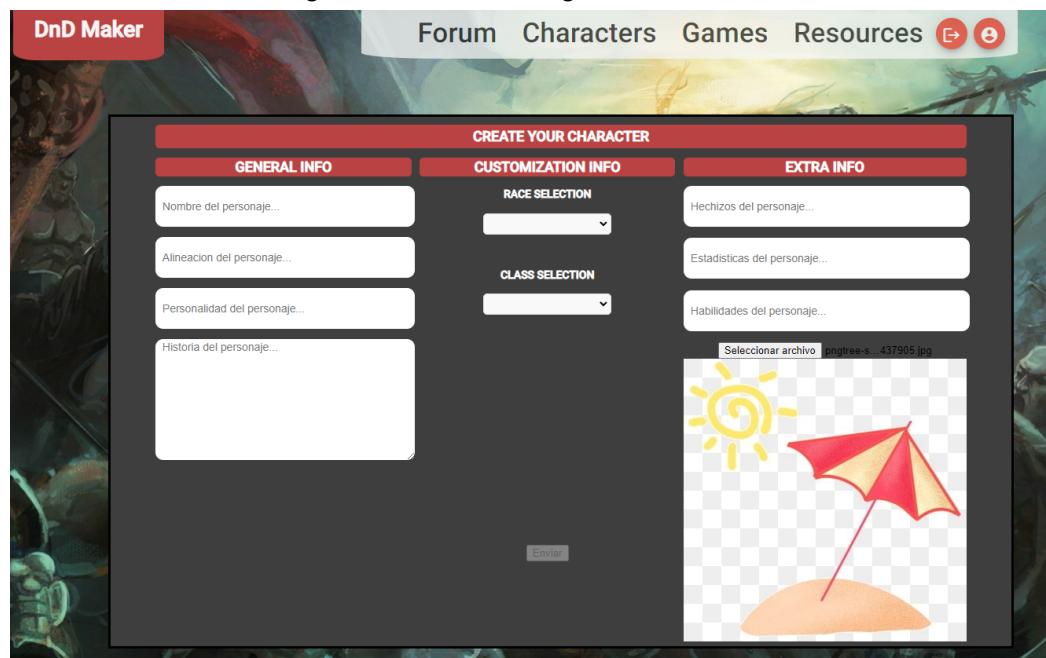
< Previous 1 2 3 4 Next >



Tendremos opción de eliminar los personajes que nos pertenezcan y se ha revisado que no se puedan eliminar los que sean de terceros.



Tendremos opción también para crear el personaje, en cuyo caso nos saldrá la siguiente interfaz. En este caso, he subido una imagen para que se pueda apreciar como se vería el formulario con una imagen subida en el angular.



Cabría a destacar la consulta que se realiza para obtener las clases y razas pickeables y es que podremos elegir tanto las que hemos creado nosotros directamente como las que han creado tercera personas y hemos importado a mano.

```
private getpickeableClasses = async (req: Request, res: Response) => {
  const valor = req.params.idOwner
  const value = req.params.array
  const array = value.split(',')
  await db.conectarBD()
  .then( async (mensaje) => {
    // const query = await ClassDB.find({$and: [{_Public: true}, {_idOwner: array}]})
    console.log(array)
    const query = await ClassDB.find({$and: [{_Public: true}, {$or: [{_id: {$in: array}}, {_idOwner: valor}]}]}))
    res.json(query)
  })
  .catch((mensaje) => {
    res.send(mensaje)
  })
}
```

3. CLASES Y RAZAS

En ambas páginas vemos dos tablas, una con nuestras creaciones y otra con las entradas generales, sin filtrar.

The screenshot shows the 'CLASSES' section of the DnD Maker website. At the top, there's a navigation bar with 'DnD Maker', 'Forum', 'Characters', 'Games', 'Resources', and two small icons. Below the navigation is a large red header bar with the word 'CLASSES'. The main content area has two tables: 'ALL CLASSES' on the left and 'YOUR CLASSES' on the right. Both tables have columns for 'NAME', 'DESCRIPTION', 'OWNER', and a settings icon. The 'ALL CLASSES' table contains several entries with names like 'asd', 'ELfo', 'Efaz', etc. The 'YOUR CLASSES' table contains entries with names like 'asd', 'ELfo', 'Efaz', 'asdasd', 'asda', 'sad', 'asd', and 'sfd'. At the bottom of each table are pagination controls ('Previous', page number, 'Next').

The screenshot shows the 'RACES' section of the DnD Maker website. At the top, there's a navigation bar with 'DnD Maker', 'Forum', 'Characters', 'Games', 'Resources', and two small icons. Below the navigation is a large red header bar with the word 'RACES'. The main content area has two tables: 'ALL RACES' on the left and 'YOUR RACES' on the right. Both tables have columns for 'NAME', 'ORIGIN', 'OWNER', and a settings icon. The 'ALL RACES' table contains entries with names like 'asdads', 'asdas', 'asd', 'asd', and 'asddsas'. The 'YOUR RACES' table contains entries with names like 'asdas', 'asd', 'asd', and 'asddsas'. At the bottom of each table are pagination controls ('Previous', page number, 'Next').

Cabría a destacar la opción de importar/exportar

| ALL CLASSES | | | |
|------------------|-------------|------------------|--|
| NAME | DESCRIPTION | OWNER | |
| sfd | dfdgsf | destroyer-335830 | |
| bea | bea | destroyer-335830 | |
| nueva | fsddsf | destroyer-335830 | |
| asdas | dsadas | assassin-734473 | |
| pruebadefinitiva | fsdjnfsd | destroyer-335830 | |

« Previous 1 2 Next »

La cual para el caso de importar funciona de la siguiente manera:

```
addClassToUser(id: number) {
  this.auth.user$
  .subscribe((profile) => {

    if(profile?.sub!==undefined) {
      let asd= this.userService.addClass(profile.sub, id)
      asd.subscribe(data => {
        console.log(data)
      })
    }
    this.ngOnInit();
  });
}
```

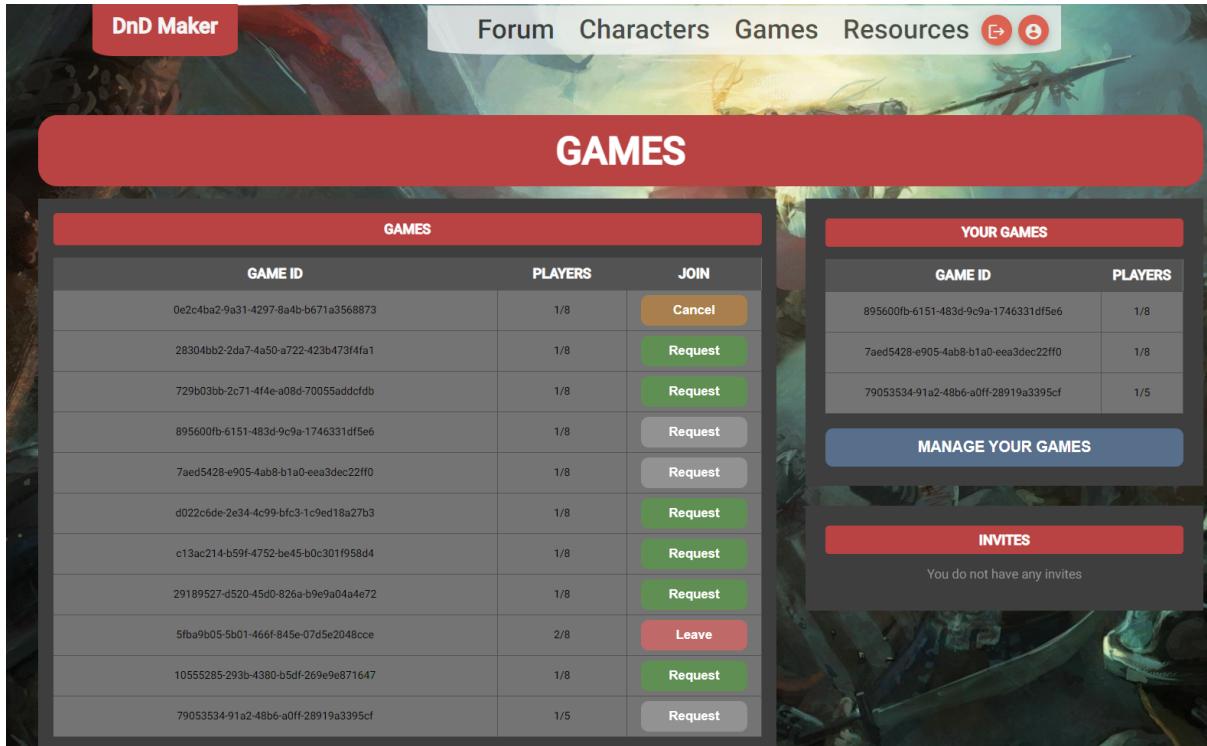
Desde html haríamos la siguiente revisión llamando al método “checkOwner” para comprobar que no nos pertenezca a nosotros la respectiva clase o raza

```
<a class="pointer" (click)="addClassToUser(Classes._id)" matTooltip="Añadir clase" style="cursor: pointer" *ngIf="checkClass(Classes._IdOwner)">
  <mat-icon>download</mat-icon>
</a>
<a class="pointer" (click)="removeClassInUser(Classes._id)" matTooltip="Añadir clase" style="cursor: pointer" *ngIf="!checkClass(Classes._IdOwner)">
  <mat-icon>clear</mat-icon>
</a>
</div>
```

4. GAMES

En el apartado de Games nos encontramos una tabla a la izquierda con todas las partidas, y a la derecha una tabla con solamente nuestras partidas en las cuales somos nosotros dueños.

Podemos ver que en función del estado de la partida respecto a nuestro usuario se dispondrá de una manera distinta y nos dejará funciones distintas.



| GAMES | | |
|--------------------------------------|---------|--------------------------|
| GAME ID | PLAYERS | JOIN |
| 0e2c4ba2-9a31-4297-8a4b-b671a3568873 | 1/8 | <button>Cancel</button> |
| 28304bb2-2da7-4a50-a722-423b473f4fa1 | 1/8 | <button>Request</button> |
| 729b03bb-2c71-4f4e-a08d-70055addcfdb | 1/8 | <button>Request</button> |
| 895600fb-6151-483d-9c9a-1746331df5e6 | 1/8 | <button>Request</button> |
| 7aed5428-e905-4ab8-b1a0-eea3dec22ff0 | 1/8 | <button>Request</button> |
| d022c6de-2e34-4c99-bfc3-1c9ed18a27b3 | 1/8 | <button>Request</button> |
| c13ac214-b59f-4752-be45-b0c301f958d4 | 1/8 | <button>Request</button> |
| 29189527-d520-45d0-826a-b9e9a04a4e72 | 1/8 | <button>Request</button> |
| 5fba9b05-5b01-466f-845e-07d5e2048cce | 2/8 | <button>Leave</button> |
| 10555285-293b-4380-b5df-269e9e871647 | 1/8 | <button>Request</button> |
| 79053534-91a2-48b6-a0ff-28919a3395cf | 1/5 | <button>Request</button> |

| YOUR GAMES | |
|--------------------------------------|---------|
| GAME ID | PLAYERS |
| 895600fb-6151-483d-9c9a-1746331df5e6 | 1/8 |
| 7aed5428-e905-4ab8-b1a0-eea3dec22ff0 | 1/8 |
| 79053534-91a2-48b6-a0ff-28919a3395cf | 1/5 |

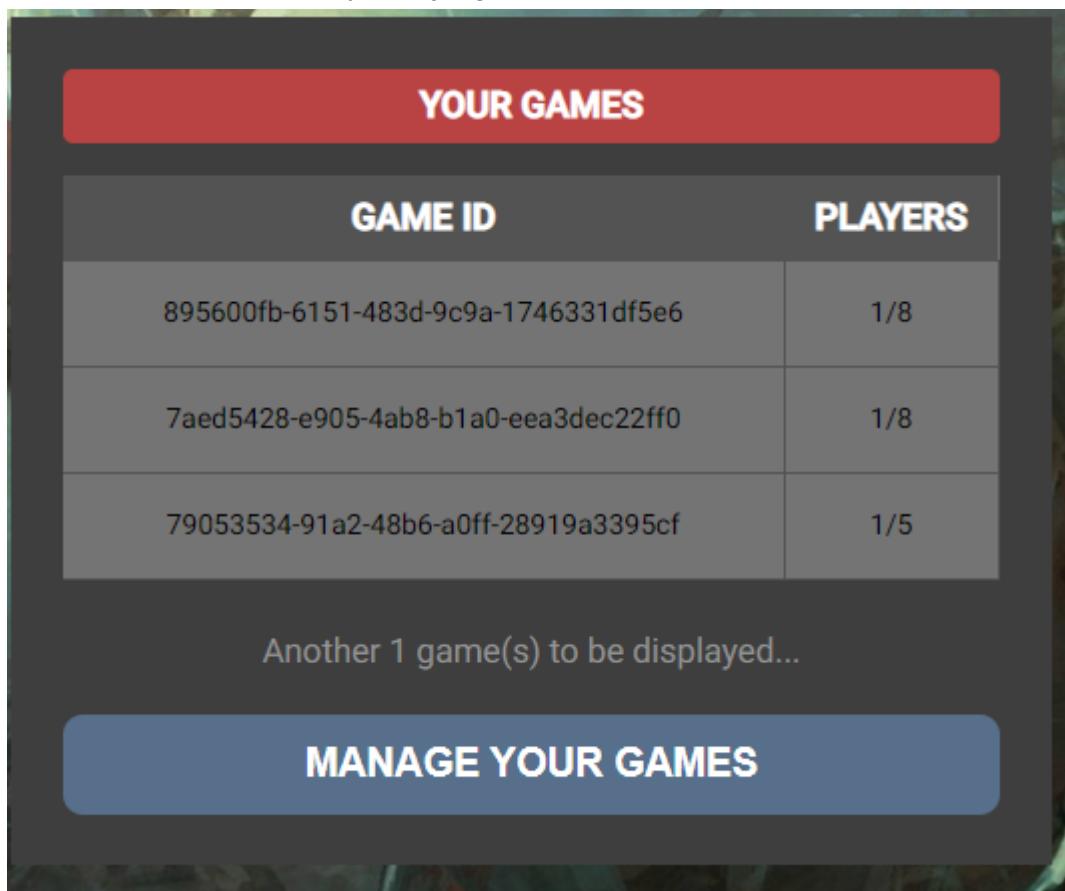
Hacemos las comprobaciones desde el código, sin tirar de memoria caché como hicimos en el foro.

Podemos ver que en función de que sea el dueño o de que formemos parte de la partida o de si ya hemos solicitado entrar nos dispondrá opciones distintas las cuales, al hacer click sobre ellas, ejecutarán distintos métodos.

```
<tr *ngFor="let game of games">
  <td>{{game._id}}</td>
  <td>{{game.participants.length}}/{{game.maxPlayers}}</td>
  <td *ngIf="auth.isAuthenticated$ | async">
    <button *ngIf="isOwner(game, profileId); else requestButtonDisabled" class="game-button-request" disabled>Request</button>
    <ng-template #requestButtonDisabled>
      <button *ngIf="existsParticipant(game); else requestButton" class="game-button-request leave"
        (click)="leaveGame(game)">Leave</button>
    </ng-template>
    <ng-template #requestButton>
      <button *ngIf="!hasPlayerRequestedJoinGame(game._id); else cancelButton" class="game-button-request">Request</button>
      <ng-template #cancelButton>
        <button class="game-button-request cancel" (click)="cancelGameRequest(game._id)">Cancel</button>
      </ng-template>
    </ng-template>
  </td>
</tr>
</table>
<ng-template #noGames>
```



En la parte de la derecha, tenemos los juegos y como podemos ver, cuando sobrepasan los 3 juegos en lugar de seguir mostrándose en pantalla a modo de lista vemos un mensaje que nos notifica de que aún hay más juegos por mostrar.

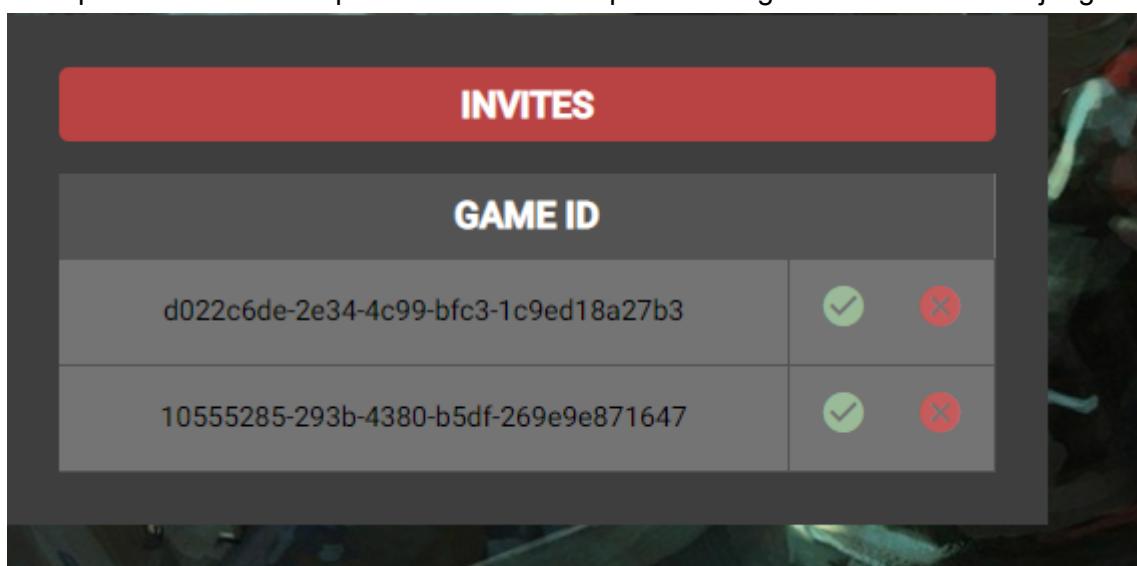


| GAME ID | PLAYERS |
|--------------------------------------|---------|
| 895600fb-6151-483d-9c9a-1746331df5e6 | 1/8 |
| 7aed5428-e905-4ab8-b1a0-eea3dec22ff0 | 1/8 |
| 79053534-91a2-48b6-a0ff-28919a3395cf | 1/5 |

Another 1 game(s) to be displayed...

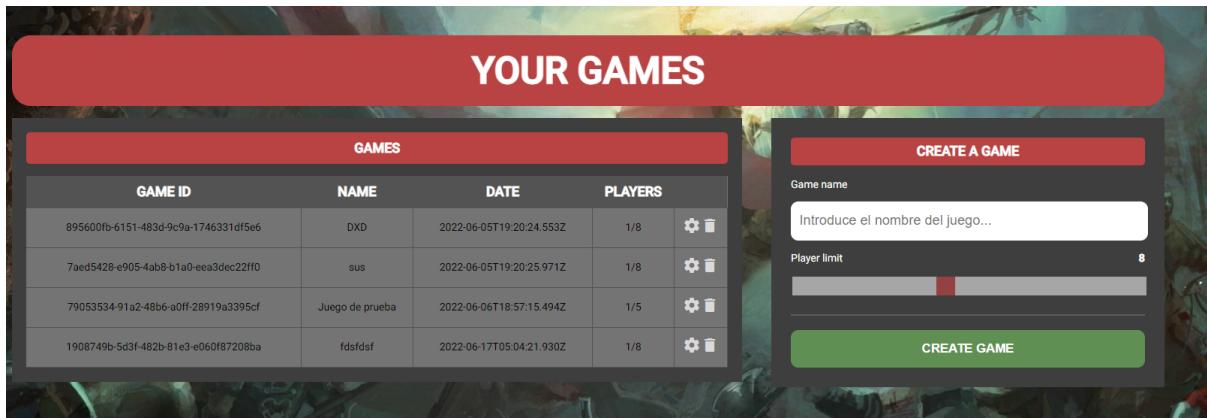
MANAGE YOUR GAMES

En la parte de “INVITES” podremos o bien aceptar o denegar las invitaciones a juego.



| GAME ID | ACCEPT | REJECT |
|--------------------------------------|--------|--------|
| d022c6de-2e34-4c99-bfc3-1c9ed18a27b3 | ✓ | ✗ |
| 10555285-293b-4380-b5df-269e9e871647 | ✓ | ✗ |

En la parte de manage podremos ver ahora sí todos nuestros juegos, teniendo a la derecha un formulario para crear un nuevo juego (Pudiendo elegir la cantidad máxima de jugadores con un slider).



YOUR GAMES

| GAMES | | | | |
|--------------------------------------|-----------------|--------------------------|---------|--|
| GAME ID | NAME | DATE | PLAYERS | |
| 895600fb-6151-483d-9c9a-1746331df5e6 | DXD | 2022-06-05T19:20:24.553Z | 1/8 | |
| 7aed5428-e905-4ab8-b1a0-eea3dec22ff0 | sus | 2022-06-05T19:20:25.971Z | 1/8 | |
| 79053534-91a2-48b6-a0ff-28919a3395cf | Juego de prueba | 2022-06-06T18:57:15.494Z | 1/5 | |
| 1908749b-5d3f-482b-81e3-e060f87208ba | fdsfdf | 2022-06-17T05:04:21.930Z | 1/8 | |

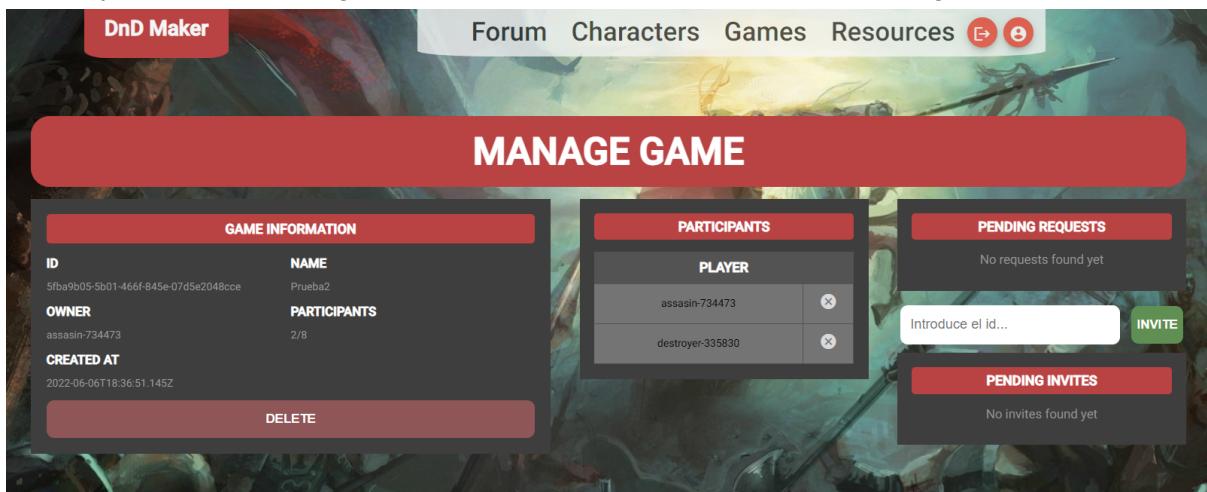
CREATE A GAME

Game name: Introduce el nombre del juego...

Player limit: 8

CREATE GAME

Si entramos a un juego podremos ver la siguiente interfaz, teniendo los datos de la partida a la izquierda, tablas para poder moderar sobre los participantes que están dentro de la partida y la capacidad de gestionar las requests e invites que se nos hagan a la partida.



DnD Maker Forum Characters Games Resources

MANAGE GAME

GAME INFORMATION

- ID: 5fb9a905-5b01-466f-845e-07d5e2048cce
- OWNER: assassin-734473
- CREATED AT: 2022-06-06T18:36:51.145Z

PARTICIPANTS

| PLAYER | |
|------------------|--|
| assassin-734473 | |
| destroyer-335830 | |

PENDING REQUESTS

No requests found yet

Introduce el id... **INVITE**

PENDING INVITES

No invites found yet



5. Áreas de mayor innovación

AUTH 0

Auth0 es una de las primeras plataformas de gestión de identidades para desarrolladores de aplicaciones y una de las pocas soluciones necesarias para las aplicaciones personalizadas.

Es utilizado como plataforma de gestión de identidades y acceso que permite a empresas de todos los tamaños administrar perfiles de usuarios, identificar problemas con los inicios de sesión y conectar aplicaciones a través de múltiples dispositivos y plataformas, entre muchas más características.

Para implementar Auth0 he tenido que agregar su extensión mediante angular

```
PS C:\Users\sonto\Documents\GitHub\DnDMaker\DnDMaker-APP> ng add @auth0/auth0-angular
i  Using package manager: npm
✓ Found compatible package version: @auth0/auth0-angular@1.9.0.
✓ Package information loaded.

The package @auth0/auth0-angular@1.9.0 will be installed and executed.
Would you like to proceed? Yes
✓ Package successfully installed.
✓ Packages installed successfully.
Nothing to be done.
PS C:\Users\sonto\Documents\GitHub\DnDMaker\DnDMaker-APP> 
```

Y luego he tenido que hacer una serie de cambios para poder utilizarlo correctamente.

En tsconfig he tenido que agregar un campo con valor true.

```
tsconfig.json () compilerOptions
1  /* To learn more about this file see: https://angular.io/config/tsconfig. */
2  {
3    "compileOnSave": false,
4    "compilerOptions": {
5      "baseUrl": "./",
6      "outDir": "./dist/out-tsc",
7      "forceConsistentCasingInFileNames": true,
8      "strict": true,
9      "noImplicitOverride": true,
10     "noPropertyAccessFromIndexSignature": true,
11     "noImplicitReturns": true,
12     "noFallthroughCasesInSwitch": true,
13     "sourceMap": true,
14     "declaration": false,
15     "downlevelIteration": true,
16     "experimentalDecorators": true,
17     "moduleResolution": "node",
18     "importHelpers": true,
19     "target": "es2017",
20     "resolveJsonModule": true, highlighted
21     "module": "es2020",
22     "lib": [
23       "es2020",
24       "dom"
25     ]
26   },

```

Y he tenido que crear un nuevo environment para auth0.

```
src > environments > environments.ts ...
1 // This file can be replaced during build by using the `fileReplacements` array.
2 // `ng build` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 import { domain, clientId } from '../../../../../auth_config.json';
6
7 export const environment = {
8   production: false,
9   auth: {
10     domain,
11     clientId,
12     redirectUri: window.location.origin,
13   },
14 };
● 14
```

No solo he tenido que hacer esas modificaciones sino he que he tenido que generar la aplicación desde auth0.com

Applications

[+ Create Application](#)

Setup a mobile, web or IoT application to use Auth0 for Authentication. [Learn more →](#)

| | | |
|--|---|---|
|  Default App Generic | Client ID: <code>zqWU6Sj3BwzUpWqhIa966TxgSJ79QXn</code>  |  |
|  DnDMaker Single Page Application | Client ID: <code>TduCsCcJ9GwxHCjqtMB80AJf6jphdM2c</code>  |  |
|  My App Single Page Application | Client ID: <code>SDM3MOIU7CSquk7LYsKe9uJtzkWgJWGS</code>  |  |



CENTRO
BILINGÜE
INGLÉS

Una vez creada, he podido recoger los valores necesarios para completar la configuración en angular

← Back to Applications

DnDMaker

Single Page Application Client ID

Quick Start Settings Addons Connections Organizations

Basic Information

Name *

Domain

Client ID

Client Secret

The Client Secret is not base64 encoded.

Description

También tuve que agregar las URL's para poder emplearlo.

[your tenants](#) / [authorize endpoint](#) [Edit more](#)

Allowed Callback URLs

```
http://localhost:4200, https://dn-dm-aker-app.vercel.app, http://alb-dndmaker-173989889.us-east-1.elb.amazonaws.com,  
https://www.dndmaker.link
```

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://`. You can use [Organization URL](#) parameters in these URLs.

Allowed Logout URLs

```
http://localhost:4200, https://dn-dm-aker-app.vercel.app, http://alb-dndmaker-173989889.us-east-1.elb.amazonaws.com,  
https://www.dndmaker.link
```

A set of URLs that are valid to redirect to after logout from Auth0. After a user logs out from Auth0 you can redirect them with the `returnTo` query parameter. The URL that you use in `returnTo` must be listed here. You can specify multiple valid URLs by comma-separating them. You can use the star symbol as a wildcard for subdomains (`*.google.com`). Query strings and hash information are not taken into account when validating these URLs. Read more about this at <https://auth0.com/docs/authenticate/login/logout>

Allowed Web Origins

```
http://localhost:4200, https://dn-dm-aker-app.vercel.app, http://alb-dndmaker-173989889.us-east-1.elb.amazonaws.com,  
https://www.dndmaker.link
```

Comma-separated list of allowed origins for use with [Cross-Origin](#)



Una vez operativo auth0 he podido llevar una contabilidad de los inicios sesión por cuenta e incluso el método que se ha utilizado para el inicio de sesión.

[Learn more →](#)

| Q. Search for users | | Search by User | X Reset |
|---|-------------------------------|----------------|----------------|
| Name | Connection | Logins | Latest Login |
| S sontor garcia sontor100@gmail.com | google-oauth2 | 15 | 10 minutes ago |
| A Adrià García Pons adriagarciapons3@gmail.com | google-oauth2 | 52 | 12 minutes ago |
| R Raúl Ruiz Reyes rruirey2301@g.educaand.es | google-oauth2 | 14 | 11 days ago |
| A Adrià García Pons agarpon2906@g.educaand.es | google-oauth2 | 1 | 21 days ago |
| A Adrià García Pons agarcia@iespuntadelverde.es | google-oauth2 | 2 | 24 days ago |
| SO sontorminecraft@gmail.com sontorminecraft@gmail.com | Username-Password-Authenti... | 1 | a month ago |

Desde el panel de auth0 tengo acceso a la siguiente información de cada usuario.

← Back to Users

sontor garcia

user_id: google-oauth2|118111497720815765491

Actions ▾

| Details | Devices | History | Raw JSON | Authorized Applications | Permissions | Roles |
|---|---------|---------|----------|---|--|-----------------------------|
| Name sontor garcia Edit | | | | Email sontor100@gmail.com (verified) | Signed Up May 10th 2022, 10:54:21 AM | |
| | | | | Primary Identity Provider Google / Gmail | Latest Login June 17th 2022, 7:08:57 AM | Accounts Associated None |
| | | | | Browser Chrome 102.0.0 / Windows 10.0.0 | | |

AWS

Amazon Web Services es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon

AWS es un proveedor de servicios en la nube, nos permite disponer de almacenamiento, recursos de computación, aplicaciones móviles, bases de datos y un largo etcétera en modalidad de cloud computing

En mi caso, a la hora de empezar con AWS tuve que crearme una cuenta gratuita (pese a que te cobren un dólar al darte de alta), y una vez con la cuenta pude comenzar a crear servicios.

The screenshot shows the AWS Billing Dashboard. On the left, there's a sidebar with navigation links like 'Inicio', 'Facturación', 'Facturas', 'Pagos', 'Créditos', 'Órdenes de compra', 'Cost & Usage Reports', 'Cost Categories', 'Etiquetas de asignación de costos', 'Free Tier', 'Billing Conductor', 'Cost Management', 'Cost Explorer', 'Budgets', 'Budgets Reports', 'Planes de ahorro', and 'Preferencias de facturación'. The main area is titled 'Panel de facturación de AWS' and shows the following data:

| Previsión total del mes actual | Saldo actual desde inicio de mes hasta hoy | Mes anterior para el mismo periodo con tendencia |
|--------------------------------|--|--|
| 6,05 USD | 6,05 USD | No hay datos para mostrar. |

Below this, it shows 'Número total de servicios activos' (1), 'Número total de cuentas de AWS activas' (1), and 'Número total de regiones de AWS activas' (1). There's also a section for 'Costo más alto' with a table:

| Nombre de servicio | Tendencia en comparación con el mes anterior | Saldo actual desde inicio de mes hasta hoy | Mes anterior para el mismo periodo |
|--------------------|--|--|------------------------------------|
| Registrar | No hay datos para mostrar. | 6,05 USD | No hay datos para mostrar. |

At the bottom right, there's a link 'Ver su factura'.

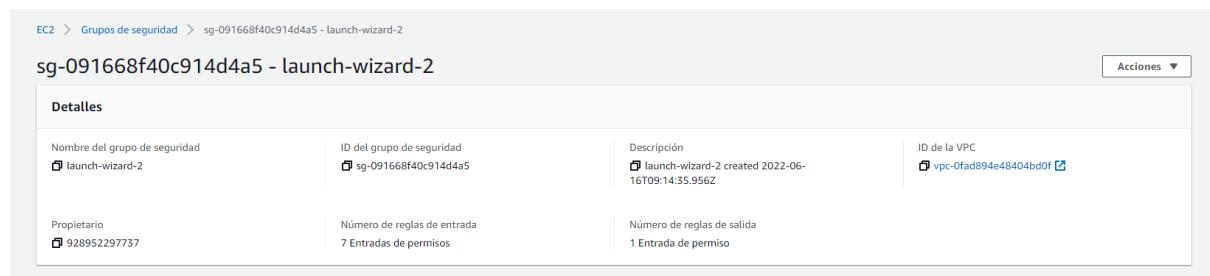
Empecé desarrollando una primera instancia pero tuve que dejarla de lado debido a que su sistema era Amazon Linux, el cual se distancia más de lo que me gustaría del Linux al que estamos acostumbrados.

Creé una segunda instancia esta vez sí en Ubuntu.

The screenshot shows the AWS Instances page with a search bar and a filter button. The main table lists two instances:

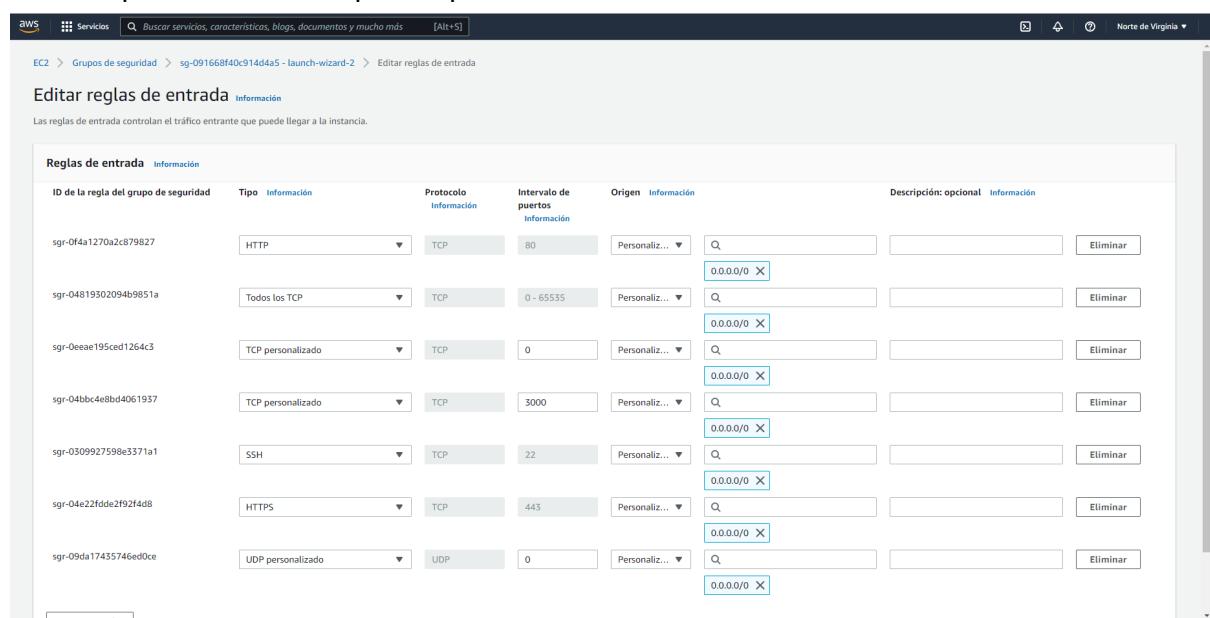
| Estado de la instancia (cliente) | ID de la instancia | Estado de la i... | Tipo de inst... | Comprobación ... | Estado de la ... | Zona de dispon... | DNS de IPv4 pública |
|----------------------------------|---------------------|-------------------|-----------------|------------------|------------------|-------------------|----------------------|
| running | i-0ca32e05687938fb5 | Detenida | t2.micro | - | Sin alarmas | us-east-1b | - |
| running | i-0f2c00086fdfb560e | En ejecución | t2.micro | 2/2 comprobador | Sin alarmas | us-east-1b | ec2-54-152-35-234.co |

En el momento de crearla me creó automáticamente un grupo de seguridad, el cuál ya he usado para el resto de servicios que me lo solicitaran.



| Nombre del grupo de seguridad | ID del grupo de seguridad | Descripción | ID de la VPC |
|-------------------------------|---------------------------|--|------------------------|
| launch-wizard-2 | sg-091668f40c914d4a5 | launch-wizard-2 created 2022-06-16T09:14:35.956Z | vpc-0fad894e48404bd0f |
| Propietario | 928952297737 | Número de reglas de entrada | 7 Entradas de permisos |
| | | Número de reglas de salida | 1 Entrada de permiso |

Este grupo de seguridad se encuentra configurado para aceptar peticiones de cierto tipo de entrada, pero de salida sí que acepta todo el tráfico saliente.



| ID de la regla del grupo de seguridad | Tipo | Protocolo | Intervalo de puertos | Origen | Descripción: opcional |
|---------------------------------------|-------------------|-----------|----------------------|---------------|-----------------------|
| sgr-0f4a1270a2c879827 | HTTP | TCP | 80 | Personaliz... | 0.0.0.0/0 |
| sgr-04819302094b9851a | Todos los TCP | TCP | 0 - 65535 | Personaliz... | 0.0.0.0/0 |
| sgr-0eeae195ced1264c3 | TCP personalizado | TCP | 0 | Personaliz... | 0.0.0.0/0 |
| sgr-04bbc4e8bd4061937 | TCP personalizado | TCP | 3000 | Personaliz... | 0.0.0.0/0 |
| sgr-0309927598e3371a1 | SSH | TCP | 22 | Personaliz... | 0.0.0.0/0 |
| sgr-04e22fdde2f92f4d8 | HTTPS | TCP | 443 | Personaliz... | 0.0.0.0/0 |
| sgr-09da174355746ed0ce | UDP personalizado | UDP | 0 | Personaliz... | 0.0.0.0/0 |

Tanto el grupo de seguridad como la instancia como el resto de servicios los he creado bajo una misma región la cual venía incluida en el paquete gratuito de AWS (us-east-1)

| | Norte de Virginia ▾ |
|------------------------------------|---------------------|
| EE.UU. Este (Norte de Virginia) | us-east-1 |
| EE.UU. Este (Ohio) | us-east-2 |
| EE.UU. Oeste (Norte de California) | us-west-1 |
| EE.UU. Oeste (Oregón) | us-west-2 |
| África (Ciudad del Cabo) | af-south-1 |
| Asia Pacífico (Hong Kong) | ap-east-1 |
| Asia Pacífico (Yakarta) | ap-southeast-3 |

Le asigné a la instancia EC3 el procesador más potente que la licencia gratuita me permitía y, tras esto, me pidió generar una serie de claves de acceso.

| | |
|--------------------------------------|---|
| Dirección IPv4 pública |  54.152.35.234 dirección abierta  |
| Estado de la instancia |  En ejecución |
| Nombre DNS de IP privada (solo IPv4) |  ip-172-31-82-113.ec2.internal |
| Tipos de instancia | t2.micro |
| ID de VPC |  vpc-0fad894e48404bd0f  |
| ID de subred |  subnet-0aa74bff8d9b7f88b  |



En mi caso generé primero de tipo “RSA” y de formato “.ppk”, ya que este formato es el que Putty tolera para llevar a cabo las conexiones ssh. Luego tuve que generar también una clave de formato .pem aunque seguía siendo tipo RSA.

EC2 > Pares de claves > Crear par de claves

Crear par de claves Información

Par de claves

Un par de claves, compuesto por una clave privada y una clave pública, es un conjunto de credenciales de seguridad que se utilizan para demostrar su identidad cuando se conecta a una instancia.

Nombre

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves | Información

RSA
 ED25519

Formato de archivo de clave privada

.pem
Para usar con OpenSSH
 .ppk
Para usar con PuTTY

Etiquetas: *opcional*

No hay etiquetas asociadas a este recurso.

[Add new tag](#)

Puede agregar hasta 50 etiquetas más.

[Cancelar](#) [Crear par de claves](#)

Tras generarlas las guardé en un directorio local y me dispuse a configurar putty para entrar a la máquina.

 dndmakerFrontKeys.ppk

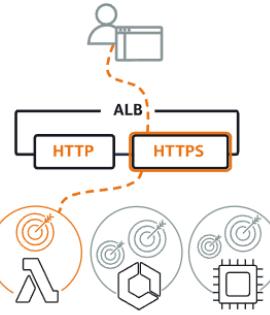
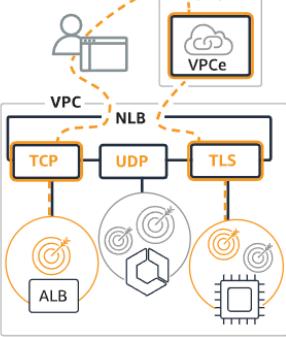
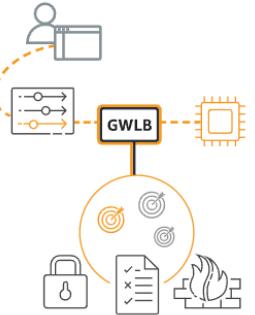


Una vez creada la instancia creé el balanceador de carga que es el componente encargado de recibir el tráfico entrante por el dns público y derivarlo al target group (el cuál en mi caso estaba formado por una sola máquina virtual).

Creé un balanceador de carga de tipo aplicación

Select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

| Load balancer types | | |
|--|---|---|
| Application Load Balancer Info  <p>Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.</p> <p>Create</p> | Network Load Balancer Info  <p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.</p> <p>Create</p> | Gateway Load Balancer Info  <p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.</p> <p>Create</p> |

Dentro del balanceador de carga lo que configuro es un agente de escucha para http tal que así:

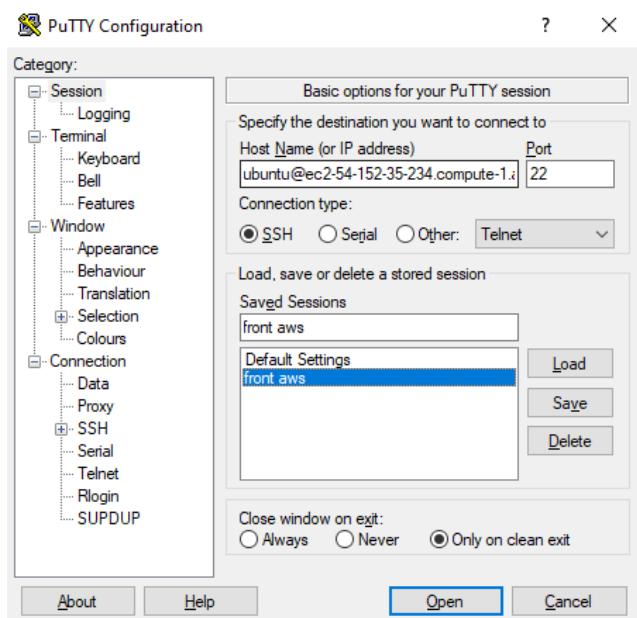
alb-dndmaker | HTTP:80 (1 reglas)

▶ Límites de regla para valores de condición, comodines y reglas totales.

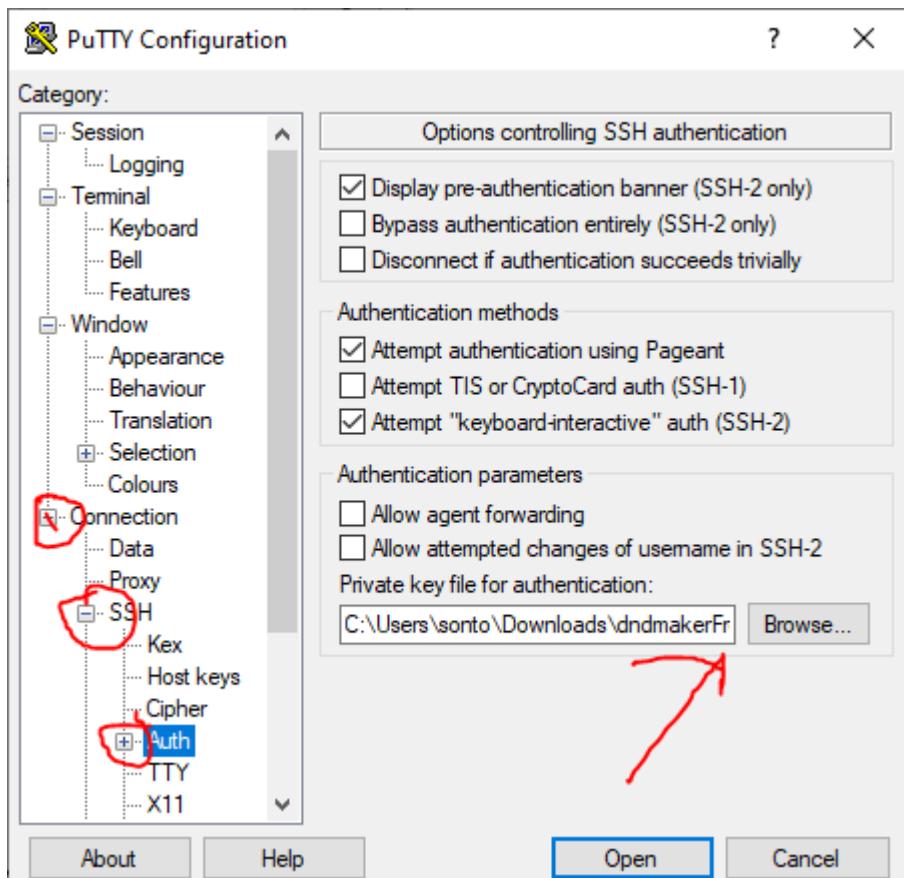
| | | |
|---|---|--|
| última HTTP 80: acción predeterminada <i>Esta regla no se puede mover ni eliminar</i> | SI ✓ Solicitudes no enrutadas de otro modo | A CONTINUACIÓN Reenviar a dndmaker-target-group-1 : 1 (100%) Persistencia de nivel de grupo: Desactivada |
|---|---|--|

El cual se conecta de la siguiente manera:

En hostname decidí poner: <usuarioMaquina>@<dnsPublicoMaquina>, el puerto 22 y conexión de tipo SSH.



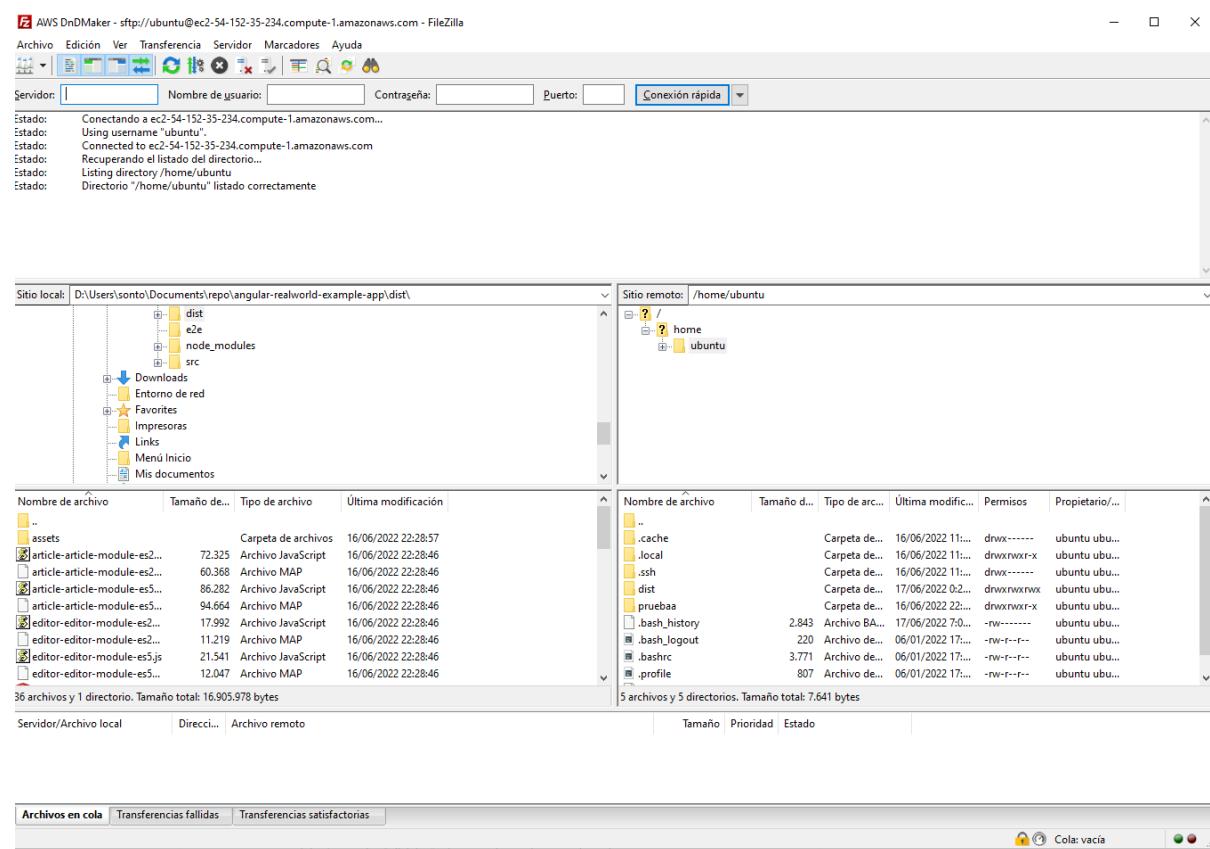
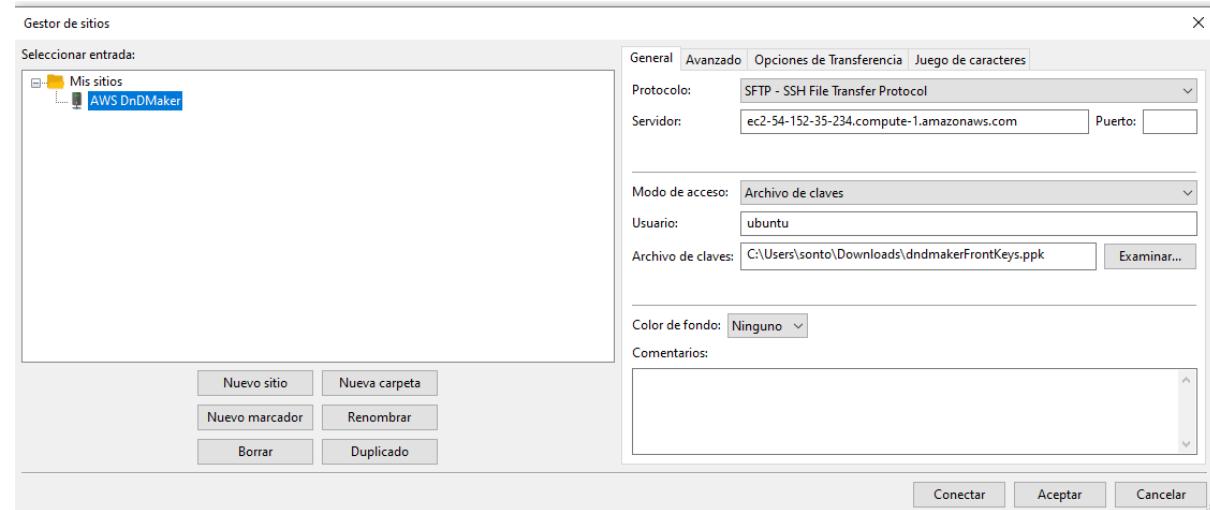
En el apartado de “Connection/SSH/Auth” adjunté el archivo .ppk que me descargué al crear la instancia.



Una vez dentro de la VM le hice un update tras lo que me dispuse a instalar todo lo necesario; nginx, node (la versión correcta), git, tsc, angular, openssl, etc.

Una vez con todo instalado traté de realizar un ng build de mi proyecto de angular pero daba igual cuanto intentase que la instancia se caía. Por lo que decidí utilizar Filezilla para pasar los archivos por FTP.

Quedando así la configuración del Filezilla



Tras hacer el ng build en mi máquina pasé los archivos a la instancia de AWS y los moví a la carpeta /var/www/html.

Tras configurar nginx para que pudiera ejecutar angular y que atendiera al directorio donde había subido los archivos me dispuse a configurar los permisos.

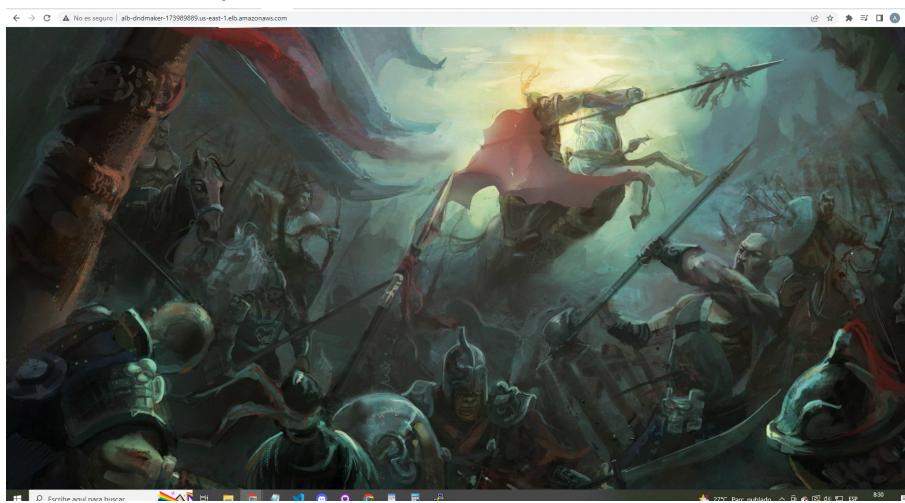
```
#root /home/ubuntu/pruebaa/dist;
root /var/www/html/dn-dmaker-app;
# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
```

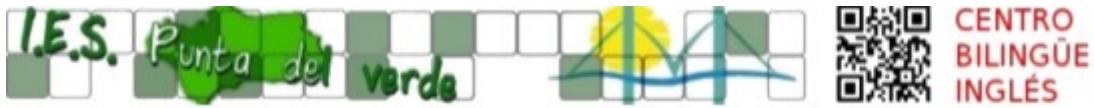
Sin embargo no importaba que hiciera que la página no me cargaba, se quedaba el fondo de pantalla congelado sin cargar nada más. Dí por hecho que nginx solo cargaba el html básico que alberga el fondo de pantalla sin ser capaz de ejecutar los archivos js que contenían mi proyecto.



Tras muchas pruebas e investigación me percaté que era problema de Auth0, el cual al no estar en un entorno https provocaba que se detuviera la ejecución.

Una vez me percaté de esto, me puse manos a la obra para intentar establecer el protocolo https como el primario de mi proyecto en aws.

Tras esto, ví que la única manera de usar https en aws era con un certificado que solo se te daba si tenías un dominio, por lo que busqué dominio gratuito pero al no encontrar ninguno que funcionara decidí comprar uno en AWS Route 53



Tras comprarlo, le creé una zona pública al dominio

Route 53 > Zonas hospedadas

| Zonas alojadas (1) | | | | | | |
|---|---------------|------------------|----------|----------|----------------------------|----------------------|
| El modo Automatic es el comportamiento actual de búsqueda que se ha optimizado para obtener los mejores resultados del filtrado. Para cambiar los modos, vaya a la configuración. | | | | | | |
| | C | Ver los detalles | Editar | Eliminar | Crear una zona alojada | |
| <input type="checkbox"/> | dndmaker.link | Pública | Route 53 | 4 | HostedZone created by R... | Z0097044BFJYC84YXBNW |

Y lo configuré para poder utilizarlo para la instancia E3.

Route 53 > Zonas hospedadas > dndmaker.link

| Pública dndmaker.link Información | | | | | | |
|---|---|---------------|---------------------------------|-------------------------------------|---|--|
| Detalles de la zona alojada | | | | | | |
| Registros (4) Información | | | | | | |
| El modo Automatic es el comportamiento actual de búsqueda que se ha optimizado para obtener los mejores resultados del filtrado. Para cambiar los modos, vaya a la configuración. | | | | | | |
| | C | Eliminar zona | Probar el registro | Configurar el registro de consultas | | |
| <input type="checkbox"/> | Registras (4) | Firma DNSSEC | Etiquetas de zona hospedada (0) | | | |
| <input type="checkbox"/> | Registros (4) Información | | | | | |
| <input type="checkbox"/> | Filtrar registros por propiedad o valor | Tipo | Política de... | Alias | | |
| <input type="checkbox"/> | Nombre del registro | Tipo | Polític... | Difer... | Valor/Dirigir el tráfico a | |
| <input type="checkbox"/> | dndmaker.link | NS | Simple | - | ns-30.awsdns-03.com. ns-1319.awsdns-36.org. ns-823.awsdns-38.net. ns-1617.awsdns-10.co.uk. | |
| <input type="checkbox"/> | dndmaker.link | SOA | Simple | - | ns-30.awsdns-03.com. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400 | |
| <input type="checkbox"/> | _c26369d3d32a1e9593c2b0628ed4c679.dndmaker.link | CNAME | Simple | - | _6b182e0b9d94e266f08b986b6f3e7cce.bpxnncpwjz.acm-validations.aws. | |
| <input type="checkbox"/> | www.dndmaker.link | A | Simple | - | 54.152.35.234 | |

Tras esto, me fui al AWS Certificate manager para solicitar mi certificado para mi dominio

AWS Certificate Manager > Certificados > Solicitar certificado

Solicitar certificado

Tipo de certificado Información

Los certificados de ACM se pueden utilizar para establecer un acceso seguro a las comunicaciones a través de Internet o dentro de una red interna. Elija el tipo de certificado que ACM debe proporcionar.

Solicitar un certificado público
Solicite a Amazon un certificado SSL/TLS público. De forma predeterminada, los navegadores y sistemas operativos confían en los certificados públicos.

Solicitar un certificado privado
No hay entidades de certificación privadas disponibles para emitir.

La solicitud de un certificado privado requiere la creación de una entidad de certificación (CA) privada. Para crear una CA privada, visite [ACM Private Certificate Authority](#)

Siguiente

Y tras una larga espera me lo otorgaron correctamente.

AWS Certificate Manager > Certificados

| Certificados (1) | | | | | | |
|--------------------------|--------------------------------------|-------------------|--------------------|--|---------------|--|
| | ID de certificado | Nombre de dominio | Tipo | Estado | ¿Está en uso? | Requisitos que se deben cumplir para la renovación |
| <input type="checkbox"/> | 4b1f7129-1fa5-4c40-9cc0-625e1256b3b2 | dndmaker.link | Emitido por Amazon | Emitido | Sí | Se cumplen los requisitos |



Por lo que pude agregar en el balanceador de carga un nuevo agente de escucha para el puerto https (443) ya que ahora sí tenía certificado operativo.

Balanceador de carga: alb-dndmaker

Descripción Agentes de escucha Monitorización Servicios integrados Etiquetas

Los agentes de escucha escuchan las solicitudes de conexión mediante su protocolo y puerto. Puede agregar, eliminar o actualizar agentes de escucha y reglas de agente de escucha.

Para ver y editar los atributos del agente de escucha, selecciónelo y elija Editar.

Agregar agente de escucha Editar Eliminar

| ID de agente de escucha | Política de seguridad | Certificado SSL | Reglas |
|--|---------------------------|--|---|
| HTTP : 80 arn...7ec31b136fc97584~ | n/a | n/a | Predeterminada: reenviando a dndmaker-target-group-1 Ver/editar las reglas |
| HTTPS : 443 arn...ec634e69b47d6b29~ | ELBSecurityPolicy-2016-08 | Predefinida: 4b1f7129-1fa5-4c40-9cc0-625e1256b3b2 (ACM) Ver/editar los certificados | Predeterminada: reenviando a dndmaker-target-group-1 Ver/editar las reglas |

Tras esto mediante openssl reconfiguré el nginx para que operara con certificados autogenerados

```
root@ip-172-31-82-113: /etc/nginx/conf.d
GNU nano 6.2
server {
    listen 80;
    listen [::]:80;
    server_name 54.152.35.234;
    return 301 https://$host$request_uri;
}

server {
    listen 443 http2 ssl;
    listen [::]:443 http2 ssl;

    server_name 54.152.35.234;

    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
    ssl_dhparam /etc/ssl/certs/dhparam.pem;

    #####
    # from https://cipherlist.eu/
    #####
    ssl_protocols TLSv1.3;# Requires nginx >= 1.13.0 else use TLSv1.2
    ssl_prefer_server_ciphers on;
    ssl_ciphers ECDH+AESGCM:EDH+AESGCM;
    ssl_ecdh_curve secp384r1; # Requires nginx >= 1.1.0
    ssl_session_timeout 10m;
    ssl_session_cache shared:SSL:10m;
    ssl_session_tickets off; # Requires nginx >= 1.5.9
    ssl_stapling on; # Requires nginx >= 1.3.7
    ssl_stapling_verify on; # Requires nginx >= 1.3.7
    resolver 8.8.8.8 8.8.4.4 valid=300s;
    resolver_timeout 5s;
    # Disable preloading HSTS for now. You can use the commented out header line that includes
    # the "preload" directive if you understand the implications.
    #add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload";
    add_header X-Frame-Options DENY;
    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection "1; mode=block";
    #####
    # END https://cipherlist.eu/ BLOCK #
    #####
}

root /var/www/html/dn-dmaker-app;

location / {
}

error_page 404 /404.html;
location = /404.html {
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
}
```



6. Conclusión

¿Qué queda pendiente?

Espero poder cumplir con los siguientes puntos pendientes durante el desarrollo del proyecto tras la entrega:

1. **Múltiples objetivos de CSS:** Mejorar la maquetación de las tablas, arreglar problemas del diseño de la navbar, mejorar el CSS del profile de usuario
2. **Mejorar la personalización del personaje:** Agregar habilidades a los personajes y, posibilidad de edición de los mismos, además, las clases y razas seleccionables deberían ser las “oficiales” + las que el usuario cree o importe.
3. **Implementar Jenkins**
4. **Aumentar la infraestructura AWS:** la API-Rest y BDD de Mongo.
5. **Securizar las sesiones mediante Tokens**
6. Seguir desarrollando el **apartado de Game** con el objetivo de que sea algo jugable
7. **Seguir implementando funciones al foro** como que puedan realizar publicaciones sobre juegos que han creado o sobre los personajes que hagan.
8. **Sistema de notificaciones** para cuando te inviten a un juego.
9. **Sistema de amigos** que salgan siempre los primeros cuando estés administrando un juego.



6. Webgrafía

Auth0:

- **Login básico Angular con Auth0:**
<https://www.youtube.com/watch?v=hWATeX1LltA&t=851s> (A raíz de lo que se hace en el vídeo pude empezar a montar el environment de Auth0)
- **Proyecto de Auth0 y angular** de “Kevin Davila Dev”
<https://github.com/kevindaviladev/Auth0Angular9-> (De aquí saqué ideas generales acerca de cómo capturar la sesión del usuario en auth0, sus datos)
- **Documentación de Auth0 oficial:**
<https://auth0.com/blog/complete-guide-to-angular-user-authentication/>
- **Login ejemplo de Auth0 con Angular:**
<https://dev.to/ricardochl/angular-login-with-auth0-4d8l>
- **Obtención del userId:** <https://community.auth0.com/t/get-userid-from-login/25802/6>
(Esto me fue muy útil y es la base de mi interacción con auth0)

Angular General:

- **Iconos de Angular:**
<https://www.angularjswiki.com/angular/angular-material-icons-list-mat-icon-list/>
<https://material.angular.io/components/icon/examples>
- **Angular Material** (En general, ya que he usado muchísimos componentes distintos):
<https://material.angular.io/>
- **CSS Botones:**
<https://stackoverflow.com/questions/45144023/angular-material-design-how-to-add-custom-button-color> (En general para los botones de los formularios)
- **Uso de botones para navegar:**
<https://stackoverflow.com/questions/47010159/how-to-redirect-to-a-new-page-in-angular-4-through-button-click> (Lo busqué para hacer la navbar)
- **Usar el fondo de pantalla con ngStyle:**
<https://stackoverflow.com/questions/34875426/how-to-add-background-image-using-ngstyle-angular2> (Lo acabé usando con ngStyle como viene en la página)



- **Para hacer la navbar:**
<https://stackoverflow.com/questions/65853465/how-do-i-get-data-from-database-to-a-drop-down-list-in-angular> (lo acabé usando)
- **Uso de ngTemplate:**
<https://medium.com/showpad-engineering/why-you-should-never-use-function-calls-in-angular-template-expressions-e1a50f9c0496> (lo acabé usando)
- **Un fallo que obtuve al compilar:**
<https://stackoverflow.com/questions/50202948/update-to-angular-v6-module-not-found-error-cant-resolve-fs>
- **Hacer referencia a funciones desde html en angular:**
<https://stackoverflow.com/questions/22723511/call-an-angular-function-inside-html>
(Lo acabé usando)
- **Dar formato a la fecha de mongo:**
<https://stackoverflow.com/questions/37597456/angular-2-and-express-date-format-mongodb> (Lo acabé usando)
- **Problema con el post:**
<https://stackoverflow.com/questions/50572363/angular-6-post-request-with-a-multipart-form-doesnt-include-the-attached-file-or>
- **La paginación que utilicé:**
<https://stackoverflow.com/questions/46558298/how-to-implement-multiple-pagination-using-ngx-pagination-in-the-same-view-in-angular>
- **Subir imágenes a angular:**
<https://stackoverflow.com/questions/47841331/upload-image-to-database-using-angular> |||| <https://www.youtube.com/watch?v=YkvqLNcJz3Y&t=323s> ||||
<https://www.youtube.com/watch?v=8GJgfk1rFUQ> ||||
<https://github.com/leifermendez/curso-angular-11/tree/main/src>
- **Problemas con el ngbuild en AWS:**
<https://techdirectarchive.com/2021/08/31/how-to-deploy-an-angular-app-to-aws-s3/> ||||
<https://blog.commutatus.com/deploy-angular-application-on-aws-elastic-beanstalk-part-1-2-cf6419728c25> ||||
<https://stackoverflow.com/questions/37558656/angular-cli-ng-build-doesnt-produce-a-working-project> ||| <https://www.youtube.com/watch?v=P213OeRoMJ4> |||
https://www.youtube.com/results?search_query=Angular+project+in+aws+ec2
- **Configuración de nginx para Angular:**
<https://faun.pub/my-nginx-configuration-for-angular-6f748a4ff683>



AWS General:

- Guía de creación de cuenta
<https://support.sou.edu/kb/articles/amazon-web-services-account-creation>
- Error con comando de instalación en la instancia EC2
<https://bobbyhadz.com/blog/aws-ec2-apt-get-command-not-found#:~:text=The%20AWS%20EC2%20%22apt%2Dget,e.g.%20sudo%20yum%20install%20curl%20.>
- Fallo de autenticación
<https://stackoverflow.com/questions/24587443/aws-disconnected-no-supported-authentication-methods-available-server-sent>
- Hacer el deploy (Pongo 1 porque el resto están en el apartado de arriba de Angular): <https://dzone.com/articles/deploying-an-angular-app-to-aws-s3>
- Mover archivos
<https://asf.alaska.edu/how-to/data-recipes/moving-files-into-and-out-of-an-aws-ec2-instance-windows/> (Para el Sftp)
- Configuración dominio
https://docs.aws.amazon.com/es_es/AmazonCloudFront/latest/DeveloperGuide/cnames-and-https-procedures.html ||||
<https://www.youtube.com/watch?v=0Mqi6BBcbjU> ||||
<https://docs.aws.amazon.com/acm/latest/userguide/setup-domain.html> ||||
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-register.html> ||||
<https://docs.aws.amazon.com/acm/latest/userguide/domain-ownership-validation.html> |||| <https://www.youtube.com/watch?v=5IfDzpkLIYY>
- Agregar SSL:
<https://stackoverflow.com/questions/40761309/adding-ssl-to-domain-hosted-on-route-53-aws>
- Crear API-Rest (Al final no lo usé):
https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/how-to-create-api.html
- Documentación general de amazon AWS (La he usado para muchas dudas varias): <https://docs.aws.amazon.com/>
- Preparación del HTTPS
<https://testcafe.io/documentation/402839/guides/advanced-guides/test-https-features-and-http2-websites>