MEMORIA PROYECTO

2ºDAW

Curso 2020

Gaston Tomas Huete

Índice

INTRODUCCIÓN

El proyecto que voy a presentar a través de esta memoria trata de una web sencilla donde poder jugar a juegos de rol de una manera no convencional es decir, poder llegar a jugar con tus amigos en un sitio web como si fuera en físico.

En primer lugar, seria conveniente explicar que es el rol o juego de rol.

‘’Un juego de rol es un juego interpretativo-narrativo en el que los jugadores asumen el ‘rol’ de personajes imaginarios a lo largo de una historia o trama en la que interpretan sus diálogos, describen sus acciones y no hay un guion a seguir, ya que el desarrollo de la historia queda por completo sujeto a las decisiones de los jugadores.”

Estos últimos meses hemos tenido que vivir una situación global muy dura por culpa del Covid-19 y que mejor forma que pasar el confinamiento que poder entretenerte con tus amigos jugando a un buen juego de rol.

El objetivo de esta pagina web es eso, poder jugar a un juego de rol en un dispositivo con conexión a internet junto a tus amigos.

DESARROLLO DEL PROYECTO

Idea de Negocio

La idea de esta web surgió un día utilizando la web/aplicación https://aminoapps.com. Puesto que me considero una persona que me gusta jugar juegos de rol vi esa aplicación muy liosa, ya que en esta puedes crear fichas pero solamente te dejan crear una en blanco. Con mi aplicación web quiero conseguir que la gente pueda jugar sin tener que diseñar todas las partes del personaje y de esta forma hacerlo mucho mas simple, ya que le proporcionas la ficha ya elaborada y de esta forma ser mas fácil su creación. Ademas de la implementación de las salas de chat donde podrás jugar con tus amigos con la máxima discreción posible sin nadie que te moleste.

Los clientes potenciales de esta aplicación son cualquier persona que le guste la temática *rol*. Con esto quiero decir que no hay una edad mínima o máxima en la que una persona pueda o no jugar al rol. Pueden ser hombres o mujeres que ya hayan jugado, que estén empezando recientemente o alguien que no haya jugado nunca. Es decir, cualquier persona, de cualquier edad o genero, puede crearse una ficha y comenzar a jugar.

Alternativas en el mercado

La alternativa a mi web que más peso tiene actualmente seria aminoapps , aplicación web que he explicado en el punto de introducción. Esta web esta enfocada a poder jugar a cualquier rol dando libertad en la creación. de dichos personajes. El problema se encuentra al querer acceder a una sala específica de juego, puesto que tienes que entrar dentro de foros y otras salas para poder llegar a la que deseas. En mi opinión esto resulta complicado y tedioso a la hora de realizar diferentes acciones dentro de la web y por ello pienso que está mal estructurada.

Por otro lado, la alternativa mas común se trataría del juego en forma presencial. Esta opción en muchas ocasiones no es viable, ya que no siempre puedes quedar con las personas que quieren jugar al rol, o en muchas ocasiones estas personas no viven en tu ciudad y es complicado desplazarse solo para desempeñar este acto.

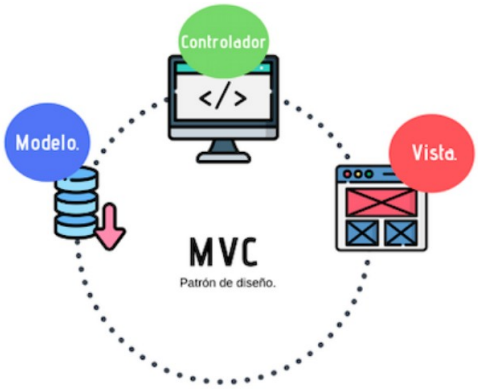
Emplear mi web para jugar al rol tiene la ventaja de que puedes seleccionar un personaje concreto de una forma sencilla y además puedes acceder a las salas donde poder jugar con otras personas en tan solo un click.

Elección de tecnologías

Estas son las tecnologías que a mi parecer han sido las correctas para desarrollar este proyecto:

- Node.js →Lo he utilizado para poder desarrollar mi aplicación y así poder montar la infraestructura de la misma.
- Handlebars → Se emplea mediante el modulo de npm llamado **express-handlebars**, es la tecnologia usada para la parte de la vista de la aplicación, es puro HTML pero con la diferencia de que puedes agregar condicionales y variables a estas vistas.
- Passport →Se trata de la tecnologia usada para la autentificación de lo usuarios que se contrasta posteriormente con la base de datos para la recuperación de estas o el registro.
- Mysql →Es la tecnologia se gasta para la persistencia, es decir, que es la parte del servidor donde se almacenan los usuarios, fichas y demases de la aplicación.
- Express →Este es el framework web utilizado para que la aplicación funcione ya que es el framework por preferencia a la hora de montar una web profesional.
- Bcrypt.js → Es un modulo de npm utilizado para encriptar y des encriptar las contraseñas mediante un algoritmo *hash* y posteriormente guardado en un sistema de persistencia.
- Connect-flash →Se trata de otro modulo de npm utilizado para poder mostrar mensajes por pantalla fácilmente.
- Bootstrap →Ademas de unos cuantos estilos realizados de manera personal, bootstrap fue mi elección ya que hace que no te compliques a la hora de dar estilo a los componentes de la aplicación.

Estructura de la aplicación



He utilizado el modelo vista controlador porque tiene la ventaja de que el código lo separas en varias partes y a la hora de hacer modificaciones o alteraciones en el mismo, personalmente me resulta mucho mas sencillo que cualquier otro método.

La estructura de la aplicación es bastante sencilla, tenemos por una parte el modelo, que son los accesos a la base de datos mediante las rutas del servidor.

```
outes > JS ficha.js > ...
const express = require('express');
const router = express.Router();
const { crear_ficha, guardar_ficha, mostrar_fichas, borrar_ficha, editar_ficha, envio_editar_ficha } = require('../controller/fichas.controller');

const { loged } = require('../lib/loged');

//fichas
router.get('/add', loged, crear_ficha);
router.post('/add', loged, guardar_ficha);
router.get('/', loged, mostrar_fichas);
router.get('/delete/:id', loged, borrar_ficha);
router.get('/edit/:id', loged, editar_ficha);
router.post('/edit/:id', loged, envio_editar_ficha);

module.exports = router;
```

A este la preceden los controladores que son los que se encargan de la funcionalidad de la aplicación, es decir, es la parte lógica de esta. Gracias a módulos como **passport y BcrypJs** podemos manejar las contraseñas de manera mas profesional al poder encriptarlas y desencriptarlas.

```
const controller_ficha = {};
const connection = require('../database');

controller_ficha.crear_ficha = (req, res) => {
    res.render('ficha/add'); //renderiza desde la direccion /add -> el fichero > /ficha/add
};

controller_ficha.guardar_ficha = async(req, res) => {

    const { nombre, alias, nivel, experiencia, fuerza, defensa, vitalidad, inteligencia, destreza, fe, carisma } = req.body;

    const inventario = crearInventario();

    const nuevaFicha = {
        nombre,
        alias,
        nivel,
        experiencia,
        fuerza,
        defensa,
        vitalidad,
        inteligencia,
        destreza,
        fe,
        carisma,
        inventario,
        user_id: req.user.id
    }
    await connection.query('INSERT INTO ficha set ?', nuevaFicha);

    req.flash('success', 'Ficha guardada satisfactoriamente'); //tiene dos parametros, nombre variable y texto

    res.redirect('/ficha');
};

controller_ficha.mostrar_fichas = async(req, res) => {
    const ficha = await connection.query('SELECT * FROM ficha WHERE user_id = ?', req.user.id);
    res.render('ficha/list', { ficha: ficha }); //renderiza el fichero list de la ruta ficha/list
```

controller > JS authentication.controller.js > ...

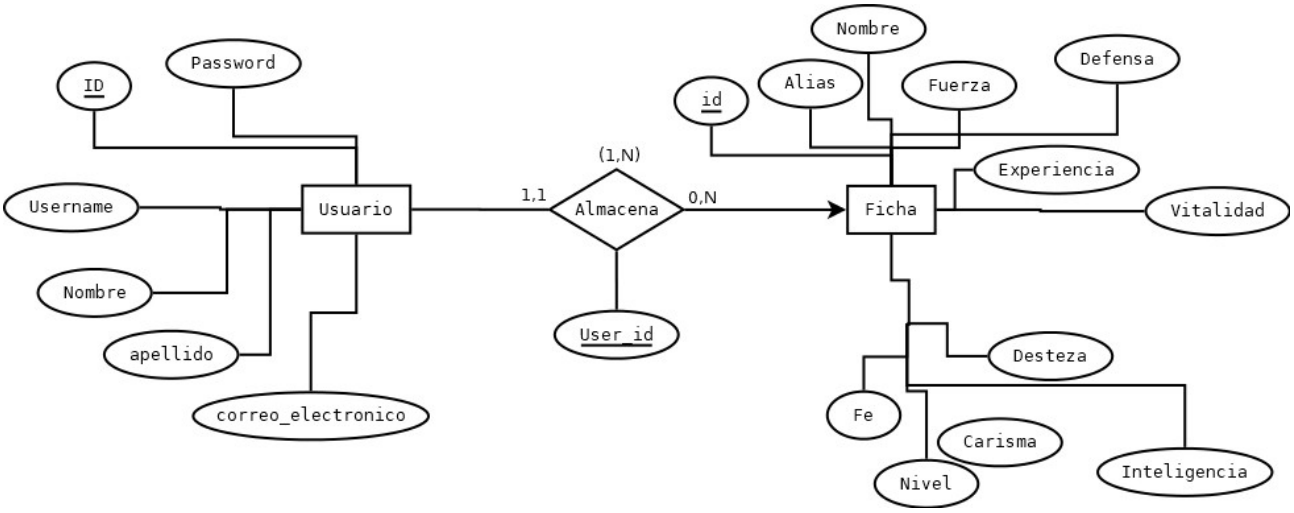
```
const controller_auth = {}
const passportLib = require('passport');

controller_auth.form_registro = (req, res) => { //la que renderiza el formulario
    res.render('auth/registro');
};

controller_auth.isSignIn = passportLib.authenticate('registro.local', { //la que envia l
    successRedirect: '/perfil',
    failureRedirect: '/registro',
    failureFlash: true
});

controller_auth.mostrar_perfil = (req, res) => {
    res.render('perfil');
};

controller_auth.form_login = (req, res) => { //la que renderiza el formulario de login
    res.render('auth/login');
};

controller_auth.login = (req, res, next) => {

    passportLib.authenticate('inicio.sesion', {

        successRedirect: '/perfil',
        failureRedirect: '/login',
        failureFlash: true
```

Y finalmente esta la vista controlada por handlebars que es la parte que vera el usuario final al utilizar la aplicación, este seria un ejemplo de la vista común echa con handlebars:

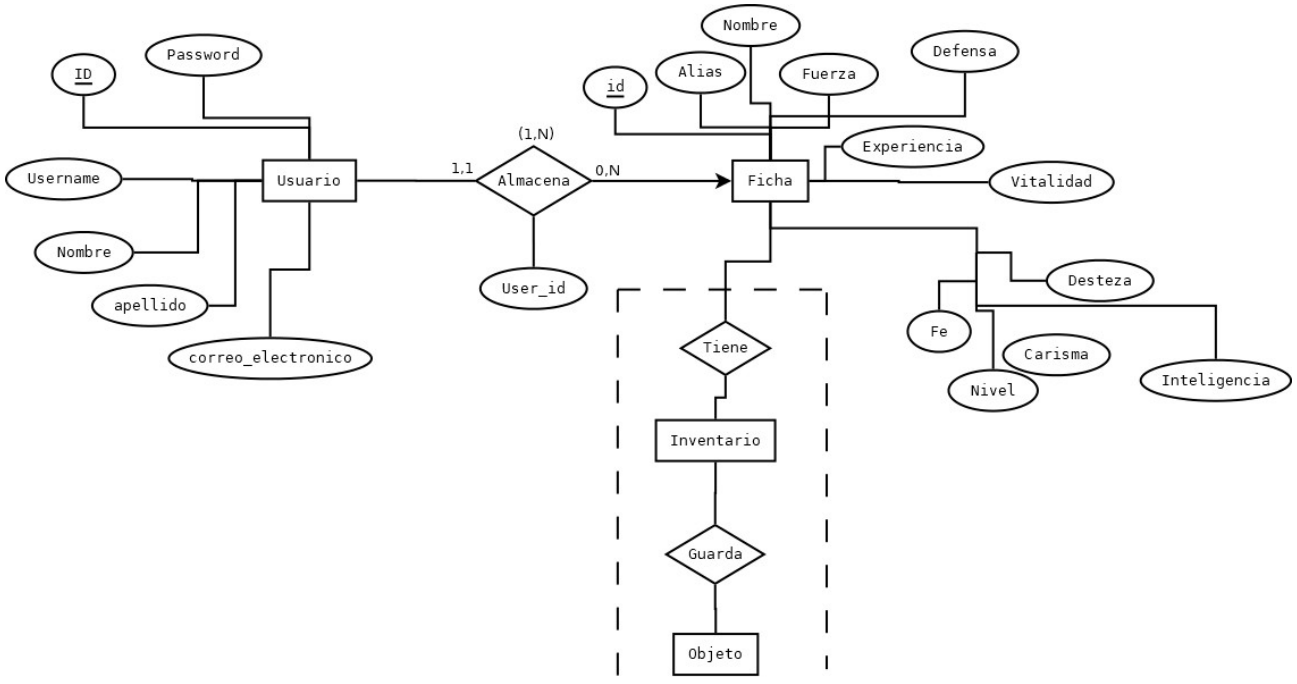
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MAGIC ROL</title>
    <!--bootstrap-->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8Wg
    <!--font awesome-->
    <script src="https://kit.fontawesome.com/c420ff5f5c.js" crossorigin="anonymous"></script>
    <!--estilo propio-->
    <style>
        @import url("/css/styles.css");
    </style>
</head>
<body>

    {{> menu }}

    {{> mensajesFlashMain}}

    {{{body}}}

    <!--scripts de bootstrap-->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anc
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfoo
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR
</body>
</html>
```

Persistencia

A continuación, muestro el esquema entidad/relación del proyecto actual:



El siguiente esquema muestra la futura actualización de la aplicación:



De esta forma se guarda la información del usuario y sus respectivas fichas.

Seguidamente el sql de la base de datos:

```
CREATE DATABASE magic_rol;

USE magic_rol;

CREATE TABLE Usuarios(
    id INT NOT NULL AUTO_INCREMENT,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    nombre VARCHAR(255) NOT NULL,
    apellido VARCHAR(255) NOT NULL,
    correo_electronico VARCHAR(255) NOT NULL UNIQUE,
    PRIMARY KEY (id)
);

CREATE TABLE ficha(
   id INT NOT NULL AUTO_INCREMENT,
   Nombre VARCHAR(255) NOT NULL,
   Alias VARCHAR(255) NOT NULL,
   Nivel INT(255) NOT NULL,
   Experiencia INT(255) NOT NULL,
   Fuerza VARCHAR(255) NOT NULL,
   Defensa VARCHAR(255) NOT NULL,
   Vitalidad VARCHAR(255) NOT NULL,
   Inteligencia VARCHAR(255) NOT NULL,
   Destreza VARCHAR(255) NOT NULL,
   Fe VARCHAR(255) NOT NULL,
   Carisma VARCHAR(255) NOT NULL,
   user_id INT,
   created_at timestamp NOT NULL DEFAULT current_timestamp,
   CONSTRAINT key_user FOREIGN KEY (user_id) REFERENCES Usuarios(id),
   PRIMARY KEY (id)
);
```

Finalmente, esta es la base de datos resultante.

```
MariaDB [magic_rol]> describe usuarios;
+--------------------+--------------+------+-----+---------+----------------+
| Field              | Type         | Null | Key | Default | Extra          |
+--------------------+--------------+------+-----+---------+----------------+
| id                 | int(11)      | NO   | PRI | NULL    | auto_increment |
| username           | varchar(255) | NO   |     | NULL    |                |
| password           | varchar(255) | NO   |     | NULL    |                |
| nombre             | varchar(255) | NO   |     | NULL    |                |
| apellido           | varchar(255) | NO   |     | NULL    |                |
| correo_electronico | varchar(255) | NO   | UNI | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.052 sec)

MariaDB [magic_rol]> describe ficha;
+-------------+--------------+------+-----+---------------------+----------------+
| Field       | Type         | Null | Key | Default             | Extra          |
+-------------+--------------+------+-----+---------------------+----------------+
| id          | int(11)      | NO   | PRI | NULL                | auto_increment |
| Nombre      | varchar(255) | NO   |     | NULL                |                |
| Alias       | varchar(255) | NO   |     | NULL                |                |
| Nivel       | int(255)     | NO   |     | NULL                |                |
| Experiencia | int(255)     | NO   |     | NULL                |                |
| Fuerza      | varchar(255) | NO   |     | NULL                |                |
| Defensa     | varchar(255) | NO   |     | NULL                |                |
| Vitalidad   | varchar(255) | NO   |     | NULL                |                |
| Inteligencia| varchar(255) | NO   |     | NULL                |                |
| Destreza    | varchar(255) | NO   |     | NULL                |                |
| Fe          | varchar(255) | NO   |     | NULL                |                |
| Carisma     | varchar(255) | NO   |     | NULL                |                |
| user_id     | int(11)      | YES  | MUL | NULL                |                |
| created_at  | timestamp    | NO   |     | current_timestamp() |                |
+-------------+--------------+------+-----+---------------------+----------------+
14 rows in set (0.061 sec)
```

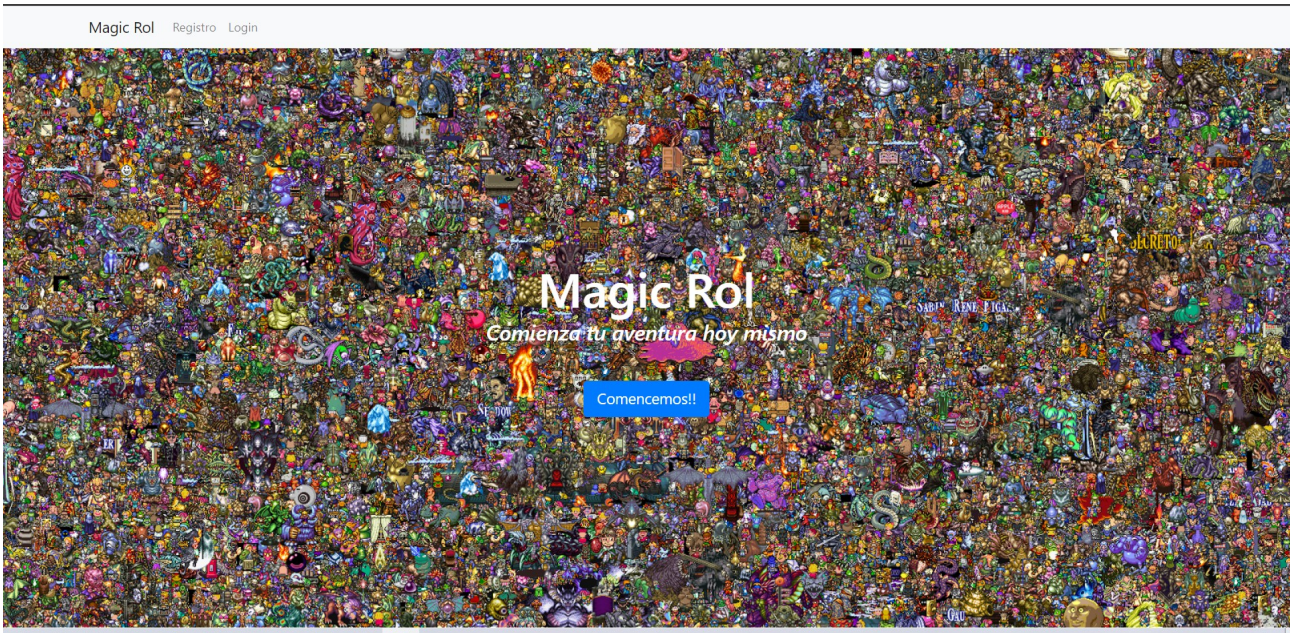
HighLights

```
const controller_auth = {}
const passportLib = require('passport');

controller_auth.form_registro = (req, res) => { //la que renderiza el formulario
    res.render('auth/registro');
};

controller_auth.isSignIn = passportLib.authenticate('registro.local', { //la que envia las cosas del formulario
    successRedirect: '/perfil',
    failureRedirect: '/registro',
    failureFlash: true
});

controller_auth.mostrar_perfil = (req, res) => {
    res.render('perfil');
};

controller_auth.form_login = (req, res) => { //la que renderiza el formulario de login
    res.render('auth/login');
};

controller_auth.login = (req, res, next) => {

    passportLib.authenticate('inicio.sesion', {

        successRedirect: '/perfil',
        failureRedirect: '/login',
        failureFlash: true

    })(req, res, next)

};

controller_auth.cerrar_sesion = (req, res) => {
    req.logOut();
    res.redirect('/login');
};
```

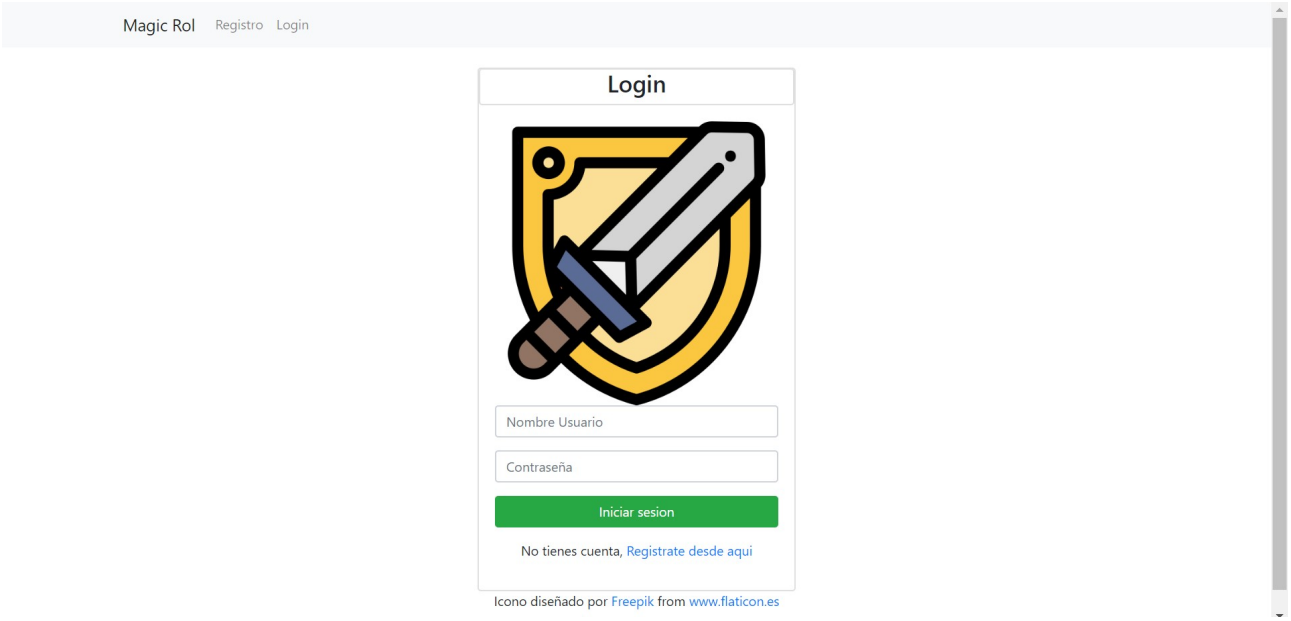
Este es uno de los fragmentos del codigo del que mas orgulloso me siento ya que controlo el inicio de sesión, el registro y también manejo si el usuario esta registrado o no.

Capturas de la aplicación en funcionamiento

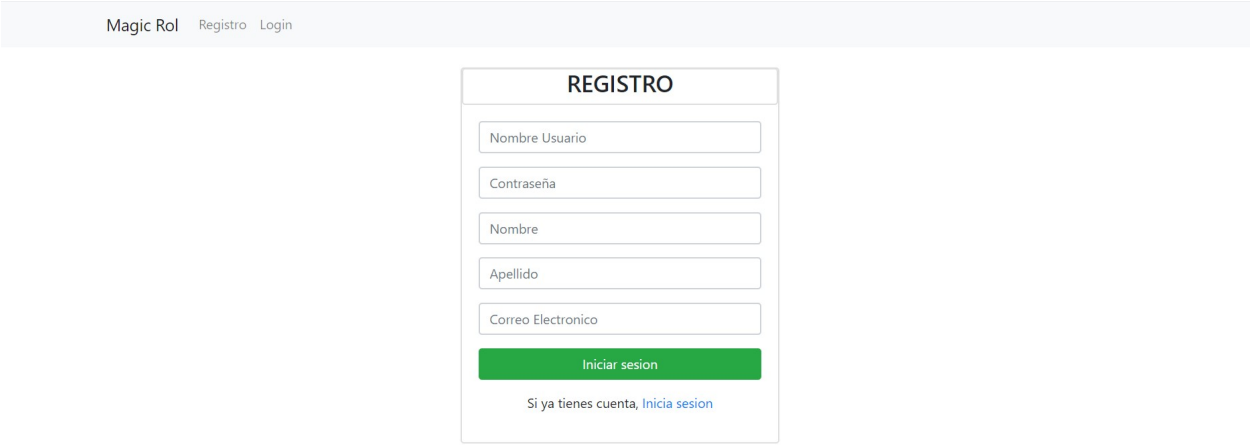
La web comienza con una pagina que nos muestra como entrar en la aplicación:



Posteriormente se encuentra el apartado del login, donde mediante un usuario y contraseña podemos acceder a la web.

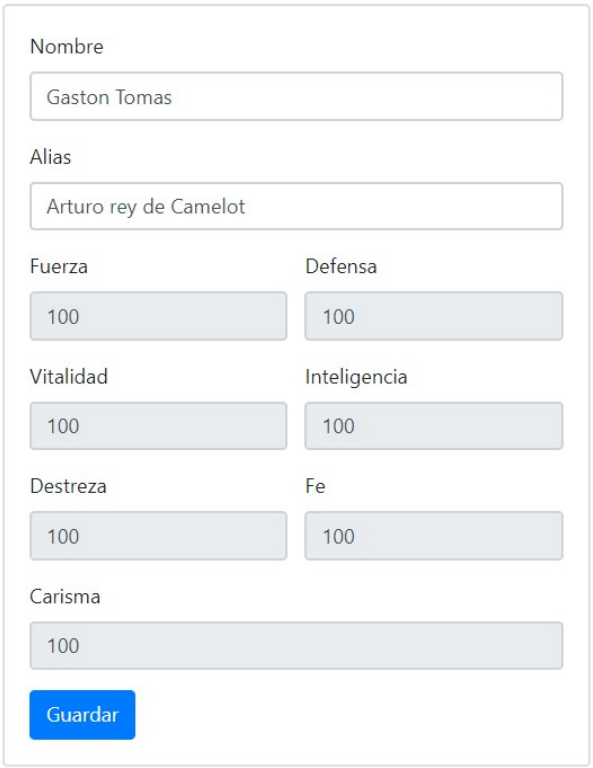


Por ende, si no tienes cuenta, puedes acceder al formulario de registro mediante el enlace del inferior de la web.



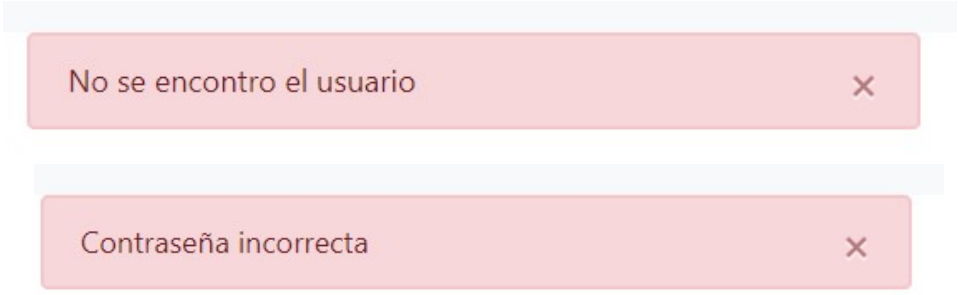
Pasado todo esto, ya estaríamos dentro de la aplicación donde nos redirecciona a nuestro perfil.

Este seria el formulario para guardar una ficha y mediante el botón de guardar lo asignaríamos a nuestro usuario.



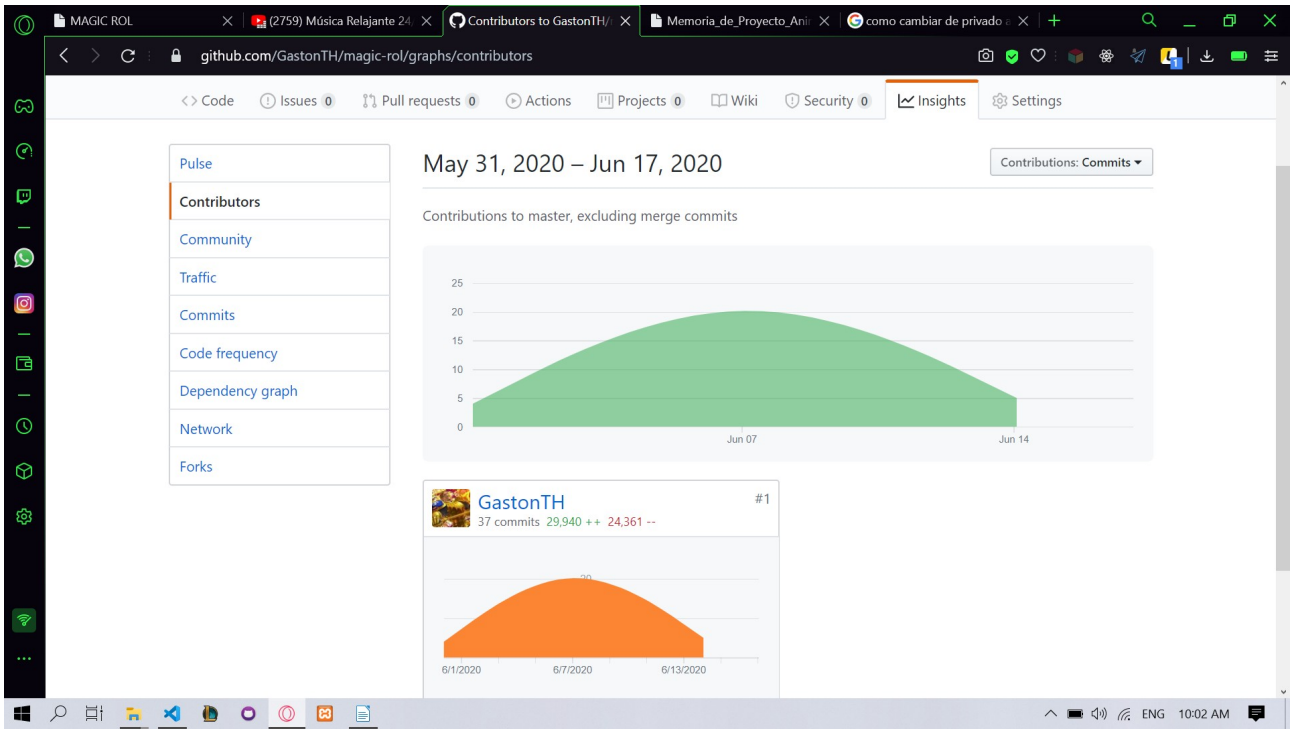
Usabilidad

- Estética y diseño minimalista.
- Facilidad de uso para el usuario medio y básico
- Ayuda a la detección de errores al registrar y al iniciar sesión.



Repositorio de Github

Este es mi repositorio de Github donde he estado subiendo el proyecto subiendo.



Y este seria el grafico de uso de github durante estos días.

Financiación y viabilidad

Para buscar financiación de la misma aplicación usaría los siguientes métodos:

- KickStarter → Es una buena forma para comenzar a ganar dinero aun siendo una aplicación gratuita.
- Concursos → Cuando la aplicación este finalizada al 100% podría presentarse a algún concurso de aplicaciones y recaudar seguidores de esta forma y ganar fama.
- Buscar patrocinadores → De esta forma la aplicación ganaría renombre y poder posicionarse en el mercado.
- Posicionamiento (SEO)

Conclusión

Yo creo que **Magic-rol** puede ser un gran proyecto ya que, en la actualidad, una web sencilla y intuitiva ayudaría mucho a la gente que empieza en este mundo de jugar al rol.

Uno de los puntos fuertes de este proyecto es la fácil ampliación del mismo por una persona cualificada, ya que el codigo es sencillo de comprender y al usar tecnologías nuevas hay bastante documentación. Cuando hablo de ampliaciones hago referencia a una sala de chat donde poder estar con gente y del mismo modo ampliar la base de datos para modificaciones como inventarios en las fichas.

Biografía

- Express (https://www.npmjs.com/package/express)
- Express-session (https://www.npmjs.com/package/express-session)
- NodeJs (https://nodejs.org/es/docs/)
- PHP (https://www.php.net/docs.php)
- Passport (http://www.passportjs.org/docs/)
- Handlebars (https://www.npmjs.com/package/express-handlebars)
- Normas APA (https://normasapa.in/)