

Ciclo Formativo de Grado Superior Desarrollo de Aplicaciones Web Curso 2020/21

Proyecto Final – Next Up Games

INDICE

Tabla de contenido

INDICE	2
INTRODUCCION.....	3
CONTENIDO.....	3
BASE DE DATOS	3
DISEÑO	4
API/SWAGGER:.....	16
CONTROL DE VERSIONES:.....	20
DESPLIEGUE DE APLICACIONES.....	20
CONCLUSIONES.....	20
BIBLIOGRAFIA.....	21

INTRODUCCION

Mi proyecto final se basa en una pagina web llamada "Next Up Games", donde uno puede estar al tanto de todos los juegos que vayan a salir en un futuro.

En la web el usuario puede registrarse para poder seleccionar los juegos que a ellos les interesan y así recibir notificaciones por email con información sobre el susodicho o recordatorios de la salida del mismo.

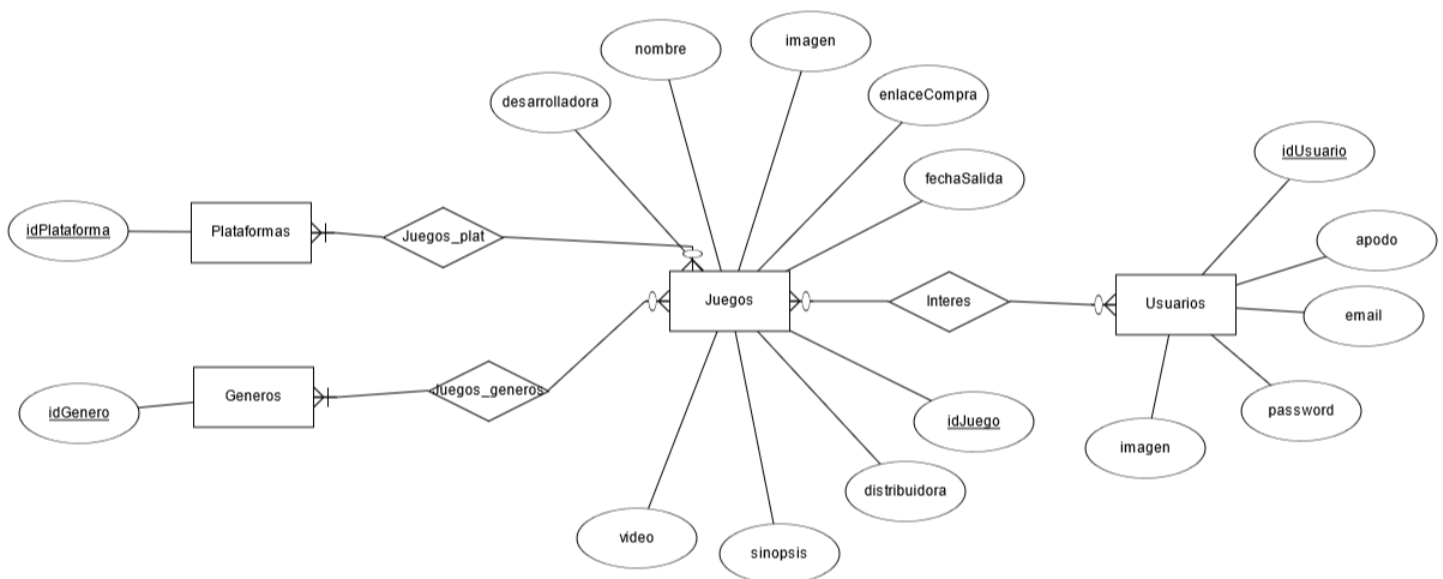
El usuario podrá buscar entre los juegos mediante el uso de nombres, géneros, fecha de salida o consolas.

Cada juego tendrá su propia pagina donde el usuario podrá ver la información del susodicho, esto incluye videos, imágenes, sinopsis, etc.

CONTENIDO

BASE DE DATOS

Lo primero que creé fueron las bases de datos. En total fueron 7 tablas, 3 de ellas siendo tablas de union.



Para el despliegue de la base de datos he usado Amazon Web Service, en concreto Amazon RDS. Una vez la base de datos estaba creada hice uso de MySQLWorkbench para conectarme y ejecutar las queries SQL.

RDS > Databases

Databases												Group resources	Modify	Actions	Restore from S3	Create database
Filter databases																
DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC	Multi-AZ						
next-up-games-db	Instance	MySQL Community	eu-west-3c	db.t2.micro	Available	3.28%	4 Connections	none	vpc-8c0e12e4	No						

MySQL WorkBench:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'NextUpGames' selected. The main window shows the 'Connection Name' tab for 'next-up-games-db'. The connection details are as follows:

- Host: ip-10-22-0-203
- Socket: /tmp/mysql.sock
- Port: 3306
- Version: 8.0.20 (Source distribution)
- Compiled For: Linux (x86_64)
- Configuration File: unknown
- Running Since: Thu Jun 17 00:41:29 2021 (0:13)

The 'Available Server Features' section shows:

- Performance Schema: Off
- Thread Pool: n/a
- Memory-based Plugins: n/a
- Semantic Replication Plugins: n/a
- SSL Availability: On

The 'Server Directories' section shows:

- Base Directory: /rdsdbbin/mysql-8.0.20.RJ/
- Data Directory: /rdsdbdata/db/
- Disk Space in Data Dir: unable to retrieve
- InnoDB Data Directory: /rdsdbdata/db/innodb
- Plugins Directory: /rdsdbbin/mysql-8.0.20.RJ/lib/plugin/
- Temp Directory: /rdsdbdata/tmp
- Error Log: On /rdsdbdata/log/error/mysql-error.log
- General Log: Off
- Slow Query Log: Off

The 'Replication Slave' section shows:

- this server is not a slave in a replication setup

The 'Authentication' section shows:

- SHA256 Password Private Key: private_key.pem
- SHA256 Password Public Key: public_key.pem

The 'SSL' section shows:

- SSL CA: /rdsdbdata/rds-metadata/ca-cert.pem
- SSL CA Path: n/a
- SSL Cert: /rdsdbdata/rds-metadata/server-cert.pem
- SSL Cipher: ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA3
- SSL CRL: n/a
- SSL CRL Path: n/a
- SSL Key: /rdsdbdata/rds-metadata/server-key.pem

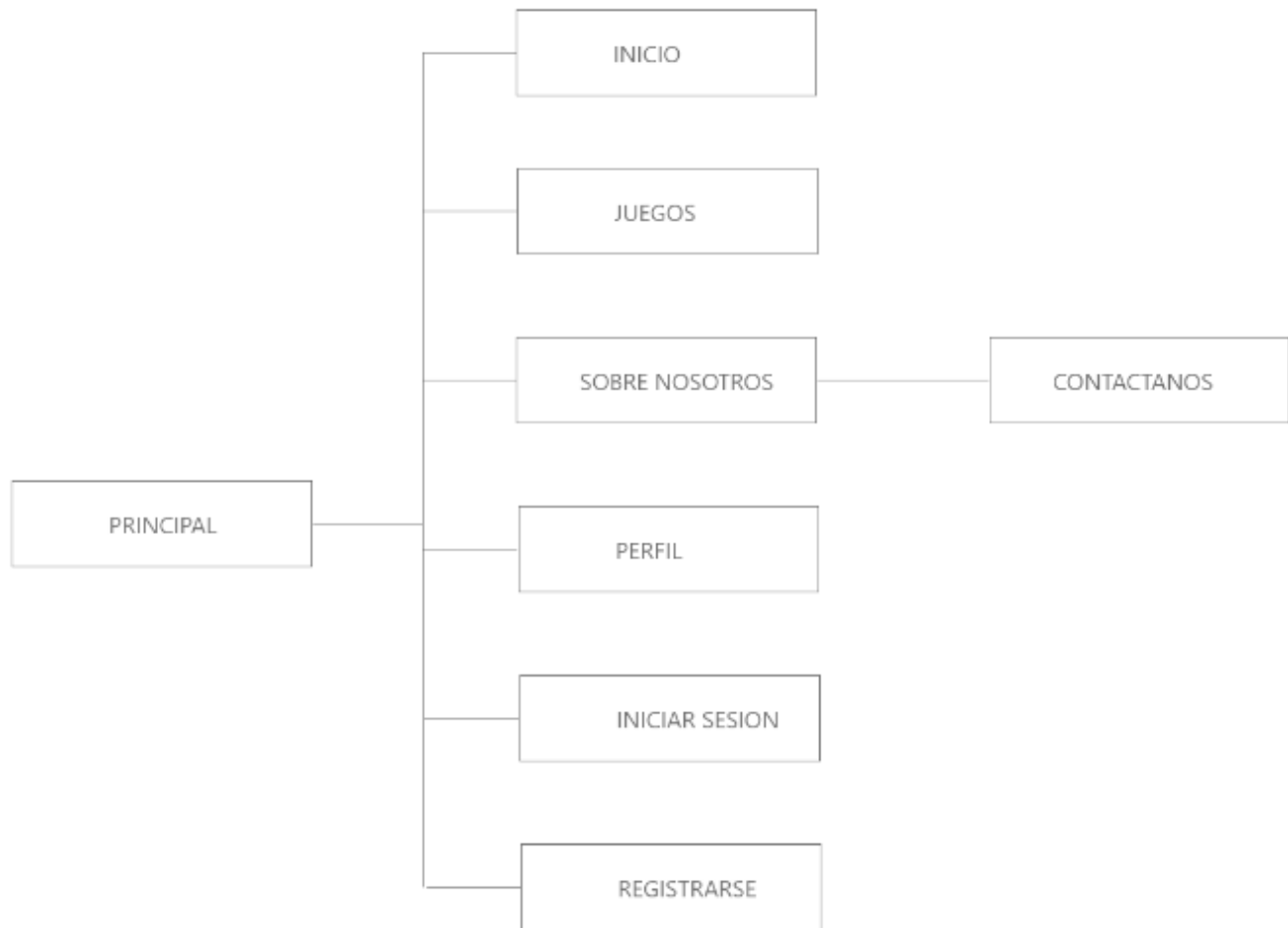
The right sidebar shows the 'Server Status' tab with the following metrics:

- Server Status: Running
- CPU Load: 3.28%
- Connections: 5
- Traffic: 4.16 KB/s
- Key Efficiency: 0.0%
- Selects per Second: 0
- InnoDB Buffer Usage: 11.3%
- InnoDB Reads per Second: 0
- InnoDB Writes per Second: 0

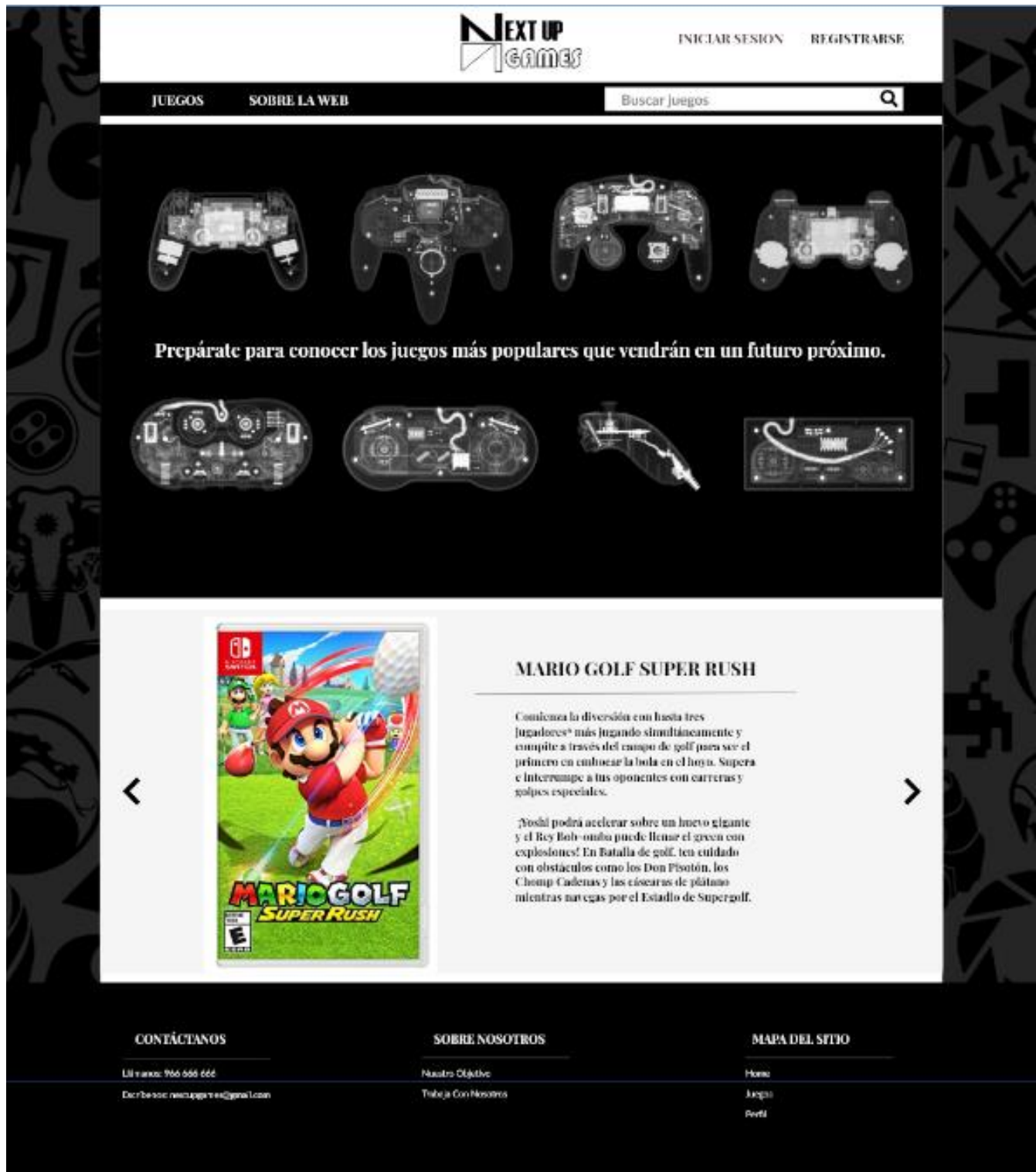
DISEÑO

Para el diseño de la página hice uso de Adobe XD. Los colores por los que me decidí fueron el blanco y el negro, que tienen un gran contraste. Como el objetivo de la aplicación es simplemente permitir a los usuarios buscar juegos, pienso que no hay mucho más que poner aparte de un poco de información sobre la web y como contactarme, por lo tanto, así concentrándome más en lo principal que son los juegos y el perfil usuario.

Mapa de Navegación:

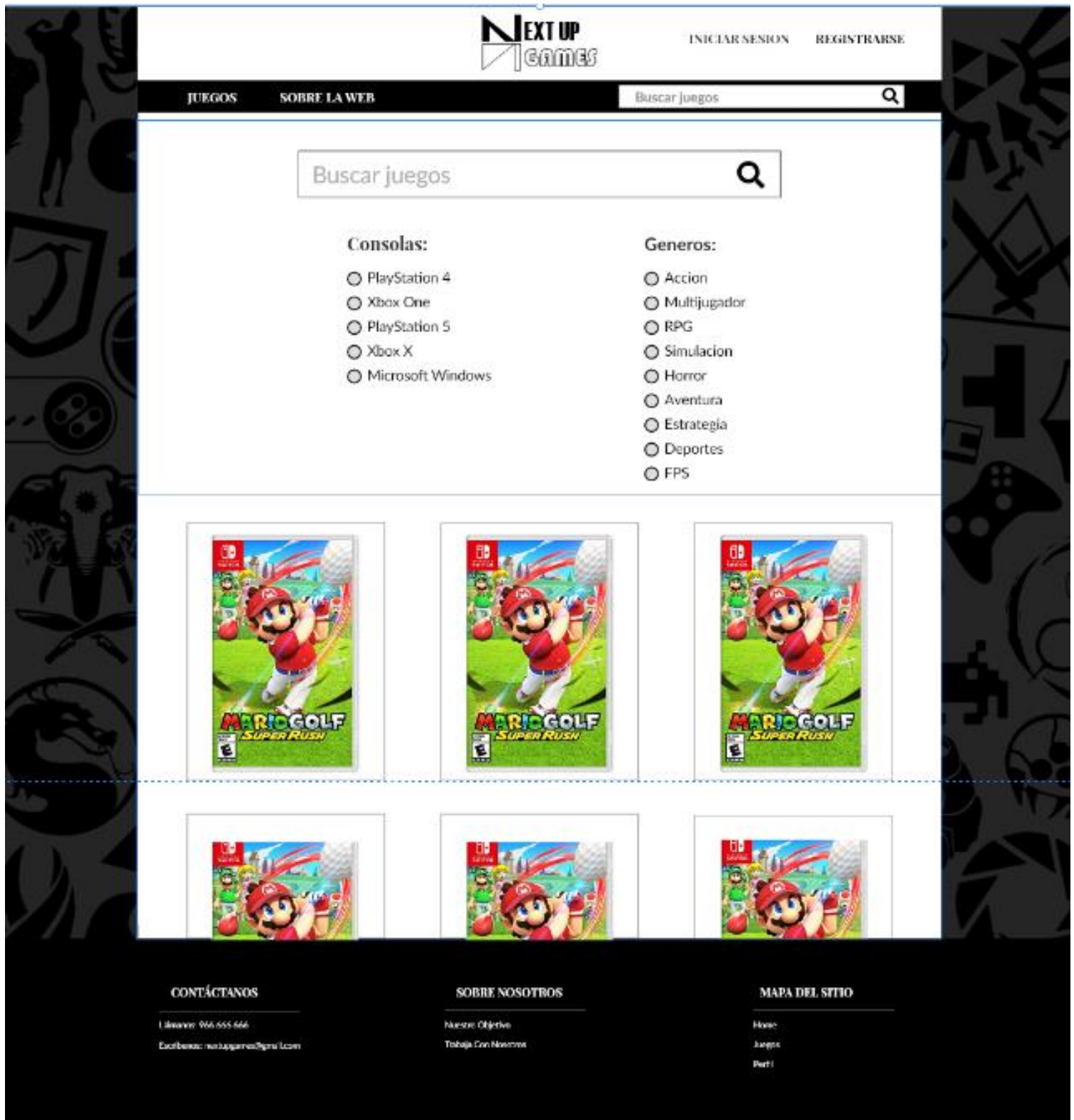


Página de Inicio:

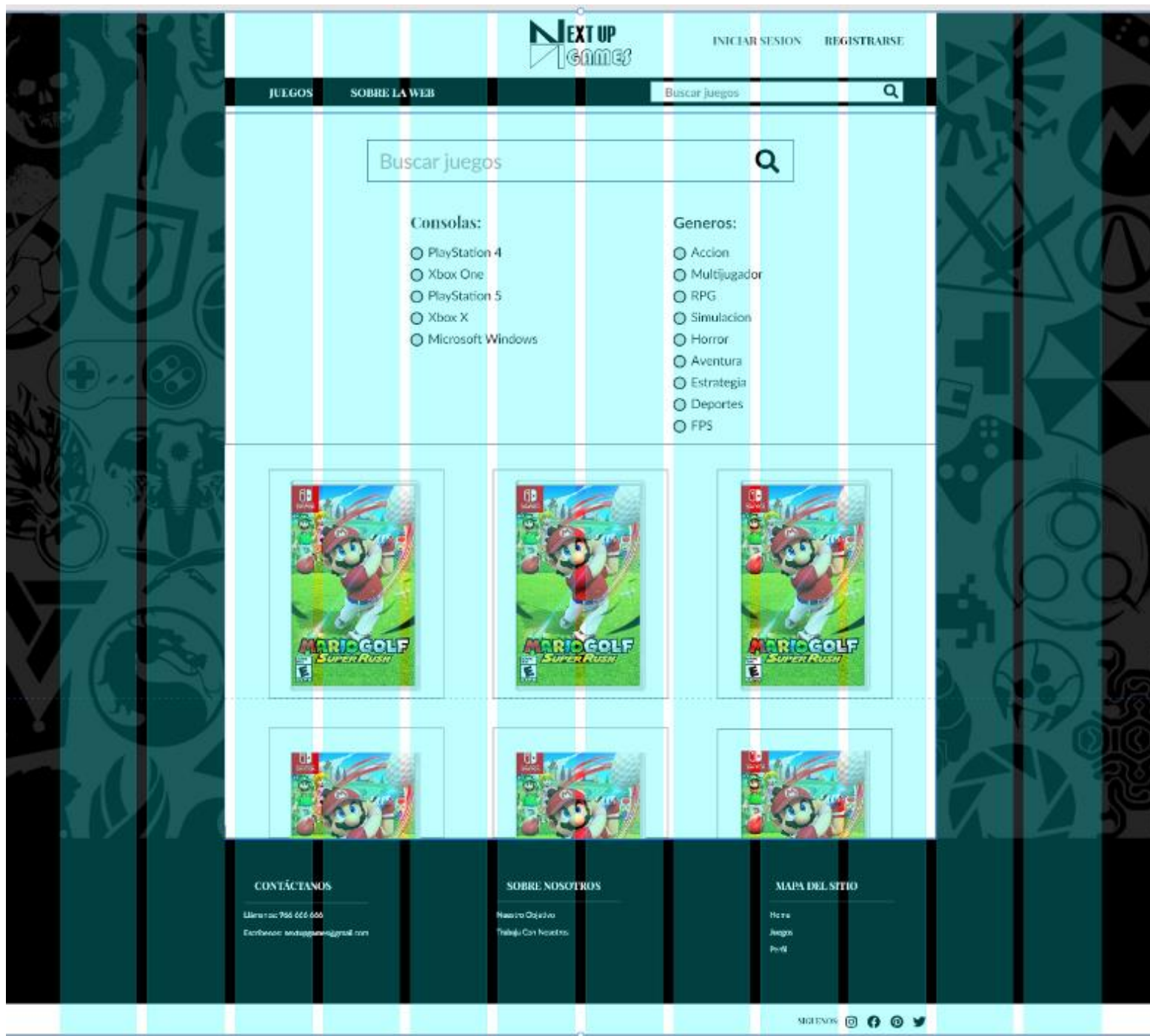


Mi objetivo era tener un diseño sencillo pero atractivo, por lo tanto, hice uso de las fuentes **Playfair Display** y **Lato** que en mi opinión combinan bastante bien.

Página de Juegos:



La cuadrícula me ha ayudado mucho para el posicionamiento de todos los componentes.



NEXT UP GAMES INICIAR SESION REGISTRARSE

JUEGOS SOBRE LA WEB

Iniciar Sesión

Email:

Contraseña:

Iniciar Sesión

[Te olvidaste la contraseña?](#) [Te olvidaste el email?](#)

Registrarse

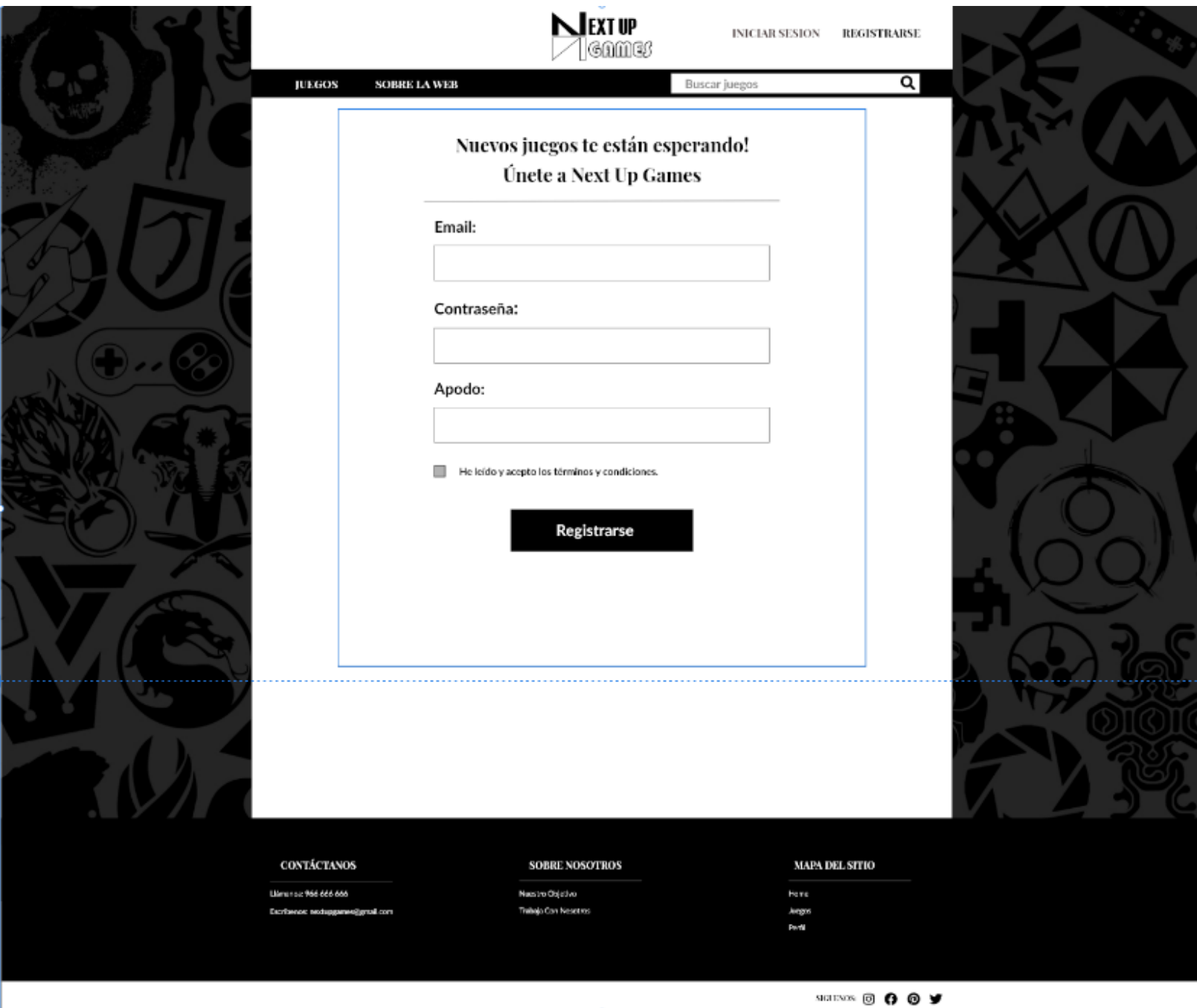
CONTÁCTANOS
Teléfono: 906 666 666
Escribenos: nextupgames@gmail.com


SOBRE NOSOTROS
Nuestro Objetivo
Trabaja Con Nosotros

MAPA DEL SITIO
[Home](#)
[Juegos](#)
[Perfil](#)

SÍGUENOS:

Página de Registro:




[INICIAR SESION](#)
[REGISTRARSE](#)

[JUEGOS](#)
[SOBRE LA WEB](#)

Nuevos juegos te están esperando!

Únete a Next Up Games

Email:

Contraseña:

Apodo:

☐ He leído y acepto los términos y condiciones.

Registrarse

CONTÁCTANOS

Llámanos al 966 666 666

Escríbenos: nextupgames@gmail.com

SOBRE NOSOTROS

Nuestro Objetivo





Trabaja Con Nosotros


MAPA DEL SITIO

Home

Acerca

Perfil

SÍGUENOS:    



[INICIAR SESION](#)
[REGISTRARSE](#)

[JUEGOS](#)
[SOBRE LA WEB](#)


MARIO GOLD SUPER RUSH

Sinopsis:

Comienza la diversión con hasta tres jugadores* más jugando simultáneamente y compite a través del campo de golf para ser el primero en embocar la bola en el hoyo. Supera e interrumpe a tus oponentes con carreras y golpes especiales.


¡Yoshi podrá acelerar sobre un huevo gigante y el Rey Bob-omba puede llenar el green con explosiones! En Batalla de golf, ten cuidado con obstáculos como los Don Písotón, los Chomp Cadenas y las cáscaras de plátano mientras navegas por el Estadio de Supergolf.

Trailer:



Plataformas: Nintendo Switch

Reservar






Fecha de Salida:
2021-06-25

Desarrolladora:
Camelot Software Planning

Distribuidora:
Nintendo

Juegos Similares:

CONTÁCTANOS

Llámanos: 966 666 666

Escríbenos: nextupgames@gmail.com





SOBRE NOSOTROS

Nuestro Objetivo


Trabaja Con Nosotros

MAPA DEL SITIO


[Home](#)
[Juegos](#)
[Perfil](#)

SÍGUENOS:    

Página de Perfil de Usuario:

INICIAR SESIONREGISTRARSE

JUEGOS SOBRE LA WEB









Perfil

Apodo:

Email:

[Cambiar Foto](#)

Juegos que te han interesado:



CONTÁCTANOS





Llámanos: 956 666 666
Escribenos: nextupgames@gmail.com

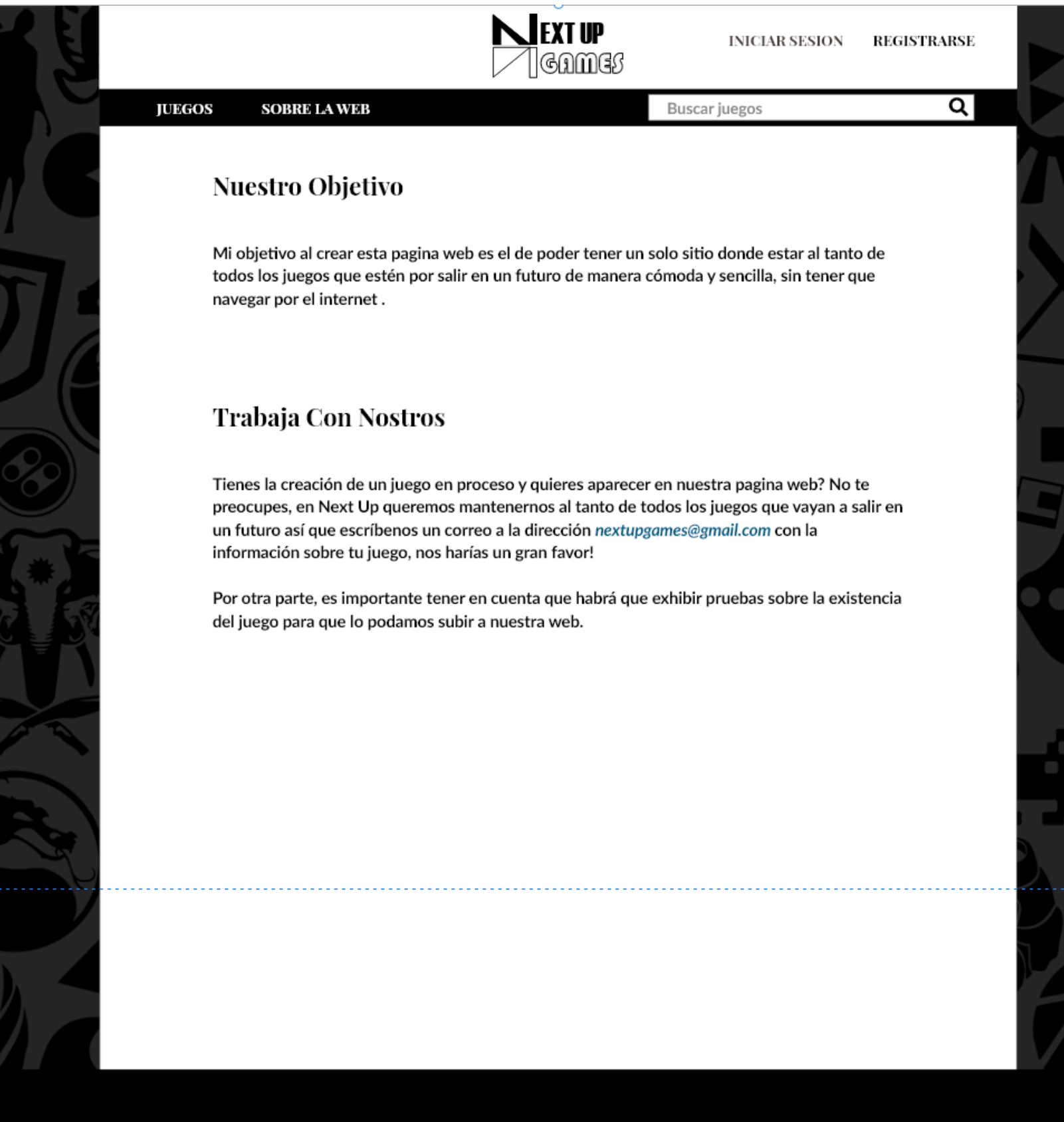
SOBRE NOSOTROS

Nuestro Objetivo
Trabaja Con Nosotros

MAPA DEL SITIO

[Home](#)
[Juegos](#)
[Perfil](#)

SIGUENOS:    





Sección Footer:



Diseño del Logo:

El diseño final acabo siendo este, uno que mezcla el boton "Next" con el titulo de la pagina. A continuacion presento el recorrido por el cual acabe decantantandome por este. Todo esto fue creado usando InkScape.



NEXT UP GAMES

LOGO

PROTOTYPES



Final->



API/SWAGGER:

Para la creación de la aplicación he comenzado usando el Swagger Editor, una herramienta que te permite generar tanto Back End como Front End asegurando un buen entendimiento entre los dos. Esto se debe a que Swagger es una serie de reglas, especificaciones y herramientas que nos ayudan a documentar nuestras APIs, y al haber sido generado por el Editor todo esto viene aplicado por defecto a ambas partes del proyecto. Esto ayuda a ahorrar tiempo en lo que ha de diseñar la API respecta. Para hacer testear la API hice uso del Postman, una aplicación gratuita que te permite hacer peticiones a tu API.

Next Up Games API 1.0.0

[Base URL: next-up-games-db.cplzfkknjtl1.eu-west-3.rds.amazonaws.com/api]

API de proyecto final Desarrollo de Aplicaciones Web.

Schemes

HTTP

usuario Operaciones sobre usuario

POST /usuario/create Crear usuario

GET /usuario/getAll Recibir todos los usuarios

GET /usuario/getOneById Recibir un usuario por id

PUT /usuario/update Actualizar usuario

DELETE /usuario/delete Borrar usuario

juego Operaciones sobre juegos

POST /juego/create Create employee

GET /juego/getAll Recibir todos los juegos

GET /juego/getOneById Recibir juego por id

PUT /juego/update Actualizar juego

DELETE /juego/delete Borrar juego

plataforma Operaciones sobre plataformas

POST /plataforma/create Crear plataforma

GET /plataforma/getAll Recibir todas las plataformas

PUT /plataforma/update Actualizar usuario

DELETE /plataforma/delete Borrar plataforma

genero Operaciones sobre generos

POST /genero/create Crear genero

GET /genero/getAll Recibir todos los generos

PUT /genero/update Actualizar genero

DELETE /genero/delete Borrar genero

interes Operaciones sobre juegos de interes, tabla de union (Juegos y Usuario)

POST /interes/create Crear interes

GET /interes/getAll Recibir todos los intereses

GET /interes/getAllById Recibir todos los intereses de un usuario por id

PUT /interes/update Actualizar interes

DELETE /interes/delete Borrar interes

juegos_genero Operaciones sobre tabla de union (Juegos y Generos) ^**POST** /juegos_genero/create Crear juegos_genero ▾**GET** /juegos_genero/getAll Recibir todos los valores de juegos_generos ▾**GET** /juegos_genero/getAllGameById Recibir todos los juegos de un genero por id ▾**GET** /juegos_genero/getAllGenreById Recibir todos los generos de un juego por id ▾**PUT** /juegos_genero/update Actualizar juegos_genero ▾**DELETE** /juegos_genero/delete Borrar juegos_genero ▾**POST** /juegos_plataforma/create Crear juegos_plataforma ▾**juegos_plataforma** Operaciones sobre tabla de union (Juegos y Plataformas) ^**GET** /juegos_plataforma/getAll Recibir todos los valores de juegos_plataforma ▾**GET** /juegos_plataforma/getAllGameById Recibir todos los juegos de una plataforma por id ▾**GET** /juegos_plataforma/getAllGenreById Recibir todas las plataformas de un juego por id ▾**PUT** /juegos_plataforma/update Actualizar juegos_plataforma ▾**DELETE** /juegos_plataforma/delete Borrar juegos_plataforma ▾**auth** Operaciones sobre autenticación ^**GET** /auth/login Logs user into the system ▾**GET** /auth/logout Hace log out a la sesion ▾**GET** /auth/recoverPassword Recibir contraseña del usuario ▾**PUT** /auth/resetPassword/{token} Recibir la contraseña del usuario ▾**PUT** /auth/changePassword Recibir contraseña de usuario ▾

Models

```
Usuario {
  idUsuario      integer($int64)
  apodo          string
  email          string($email)
  password       string
  imagen         string
}
```

```
Juego {
  idJuego      integer($int64)
  nombre       string
  imagen       string
  fechaSalida  string($full-date)
  sinopsis     string
  desarrolladora string
  distribuidora string
  video        string
  enlaceCompra string
}
```

```
Plataforma {
  idPlataforma string
}
```

```
Genero {
  idGenero string
}
```

```
Interes {
  idJuego      Juego/properties/idJuego integer($int64)
  idUsuario    Usuario/properties/idUsuario integer($int64)
}
```

```
Juegos_genero {
  idJuego      Juego/properties/idJuego integer($int64)
  idGenero     Genero/properties/idGenero string
}
```

```
Juegos_plataforma {
  idJuego      Juego/properties/idJuego integer($int64)
  idPlataforma Plataforma/properties/idPlataforma string
}
```

FRONTEND

For the FrontEnd y used Angular with TypeScript. Through swagger I was able to generate all the services and models required for the FrontEnd, and with some mild tinkering it was already good to go. I decided to make the web page narrow, as opposed to normal webpages that occupy all of the screen. My inspiration for this came from the webpage MyAnimeList, a site which I have always found to have a very nice flow to its design. For that I created in the routing file a component that would be the “father” of the other components, by doing this whenever a “child” component was to change the “father” component will always remain still and in place.

CONTROL DE VERSIONES:

Para el control de versiones he hecho uso del Git y GitHub. El proyecto lo tengo dividido en dos repositorios, uno para el BackEnd y otro para el FrontEnd. Cada tiene dos ramas distintas, la main y una llamada local, que es donde me encuentro la mayoría del tiempo programando. Una vez funciona todo bien, procedo con un merge entre las dos ramas y subo los cambios al repositorio remoto en GitHub.

Repositorio BackEnd:

<https://github.com/Pabarino/BackProyecto>

Repositorio FrontEnd:

<https://github.com/Pabarino/FrontProyecto/tree/main>

DESPLIEGUE DE APLICACIONES

Tanto el BackEnd como el FrontEnd los he desplegado en Heroku, una página web gratuita y relativamente fácil de usar. El despliegue lo hace enteramente la pagina web, y como usuario tienes que seleccionar el repositorio con la aplicación que quieres desplegar. Aun así, he tenido problemas realizando una conexión entre mi API y la base de datos en AWS RDS. Esto se debe a que RDS bloquea la conexión de Heroku y por mucho que manipulara las reglas de entrada y salida al servidor, llegando a permitir la entrada de cualquier IP, no había conexión alguna entre Heroku y AWS. Debido a esto he realizado la parte de la API en local ya que esta si conecta con el servidor RDS.

Para desplegar en Heroku he hecho uso del Heroku cli, un programa que te instalas y te permite subir y desplegar directamente desde la Git Bash. Esto se debe a que al desplegar directamente desde mi repositorio GitHub las páginas web no recibían conexión.

CONCLUSIONES

El proyecto me ha parecido a la vez complicado como gratificante, especialmente los momentos en los que finalmente solucionaba algún problema que me había estado fallando durante horas o días. Creo que he aprendido mucho a lo largo de este trabajo individual, en concreto haber puesto todo lo aprendido este curso en un solo proyecto me ha abierto los ojos a varias cosas que no me acababan de aclarar a lo largo del año.

Después de haber estado trabajando en equipo a lo largo de la FCT, al tener que trabajar solo he notado mucho mas el esfuerzo y trabajo que hay detrás una aplicación web.

BIBLIOGRAFIA

SWAGGER:

<https://editor.swagger.io/#>

AYUDA EN DESPLIEGUE Y CONEXIÓN A BASE DE DATOS:

<https://aws.amazon.com/es/getting-started/hands-on/create-mysql-db/>

IMÁGENES DE JUEGOS:

<https://www.amazon.es>

<https://www.game.es>

IMÁGEN DE PERFIL VACIO:

https://www.worldfuturecouncil.org/p/helen-clark-3/blank-profile-picture-973460_1280/