



MHRise Optimizer

Luis Rafael Arroyo Caballero



Título del proyecto: MHRise Optimizer
Alumno: Luis Rafael Arroyo Caballero
Curso académico: 2022-2023
Ciclo Formativo: 2º Desarrollo de Aplicaciones Web



Indice

Introducción.....	3
Identificación de las necesidades del proyecto.....	3
Breve análisis con las alternativas del mercado.....	4
Justificación del proyecto.....	4
Stack tecnológico.....	5
Modelo de Datos.....	7
Prototipo de la Aplicación Web.....	9
Definición API REST.....	11
Manual de Despliegue.....	11
Postmortem y conclusiones del proyecto.....	18



Introducción

El conocido videojuego Monster Hunter Rise tiene una gran cantidad de jugadores que buscan llevar su personaje al siguiente nivel a medida que progresa en su aventura. Por desgracia, para la mayoría de jugadores es un proceso bastante tedioso y confuso, pues las herramientas que proporciona el juego para optimizar tu personaje son pocas e insuficientes.

Por ello la aplicación MHRise Optimizer busca ofrecer a los jugadores una herramienta para tener una experiencia mucho más llevadera a la hora de progresar en tu aventura, con herramientas y enlaces de interés que permitirán a los usuarios llevar un seguimiento de los principales objetivos a superar para mejorar tu personaje. Además, nuestra aplicación contiene gran cantidad de información sobre el juego, tanto para jugadores menos experimentados que no buscan sobrecargarse de información como para jugadores avanzados cuyo objetivo es la perfección de su personaje.

Identificación de las necesidades del proyecto

- RF01. Permitir registrarse e iniciar sesión a los usuarios de la aplicación.
- RF02. Permitir a los usuarios cerrar sesión.
- RF03. Los usuarios podrán añadir armaduras a su inventario personal.
- RF04. Los usuarios podrán consultar las armaduras añadidas al arsenal.
- RF05. Los administradores podrán añadir, consultar, borrar y modificar usuarios.
- RF06. Se mostrarán los datos de los monstruos a los usuarios.
- RF07. Se mostrará la lista de armaduras a los usuarios.
- RF08. La aplicación mostrara enlaces de interés para los usuarios.



Breve análisis con las alternativas del mercado

A día de hoy, como en la mayoría de videojuegos tan solo tenemos diferentes Wikis hechas por seguidores de la comunidad las cuales contienen la mayor parte de información del juego. Aún teniendo en cuenta la gran cantidad de datos que proporcionan estas aplicaciones, no poseen ningún sistema de seguimiento de materiales o armamento para facilitar al jugador los objetivos que se debe de marcar para obtener las diferentes armas o armaduras.

Debido a ello, esta aplicación web puede llegar a tener un gran impacto en la comunidad destacando frente a otras que tan solo proporcionan información sobre los monstruos y equipamiento.

Sin embargo, nuestra página no contiene información tan detallada como podrían tener estas Wikis mencionadas anteriormente, por lo que es probable que sigan mostrando una alta competencia a nivel de información.

Justificación del proyecto

El motivo principal que ha dado lugar a comenzar este proyecto fue el gran seguimiento e impacto que ha tenido esta saga a lo largo de los años, tanto en el mundo de los videojuegos como a nivel personal. Al haber estado en contacto tanto con las diferentes entregas como con la comunidad, salieron a relucir ciertas carencias que estas entregas no ofrecían y que la comunidad cree necesarias.

Debido a ello surgió la idea de este proyecto, con el fin de suplir esa falta de ayuda al jugador a la hora de armar al personaje, ofreciendo un sistema con el que realizar un seguimiento de forma sencilla tanto del equipamiento deseado como de los materiales necesarios para su creación.



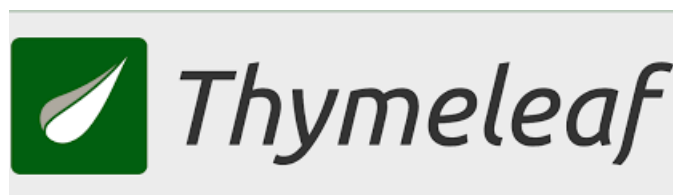
Stack tecnológico



Spring boot: Framework de código abierto que da soporte para el desarrollo de aplicaciones y páginas webs basadas en Java. Entorno de desarrollo popular que ayuda a los desarrolladores back-end a crear aplicaciones de alto rendimiento empleando objetos de java sencillos



SQL (Structured Query Language): Lenguaje de programación que permite manipular y descargar información de una base de datos. Utilizado en este proyecto debido a que se integra bien con los diferentes lenguajes de programación, entre ellos Java, utilizado en el desarrollo de esta aplicación.



Thymeleaf: Tecnología que nos permite definir una plantilla junto con un modelo de datos para obtener un nuevo documento. Utilizado para facilitar la integración del back-end y el front-end de la aplicación.



HTML



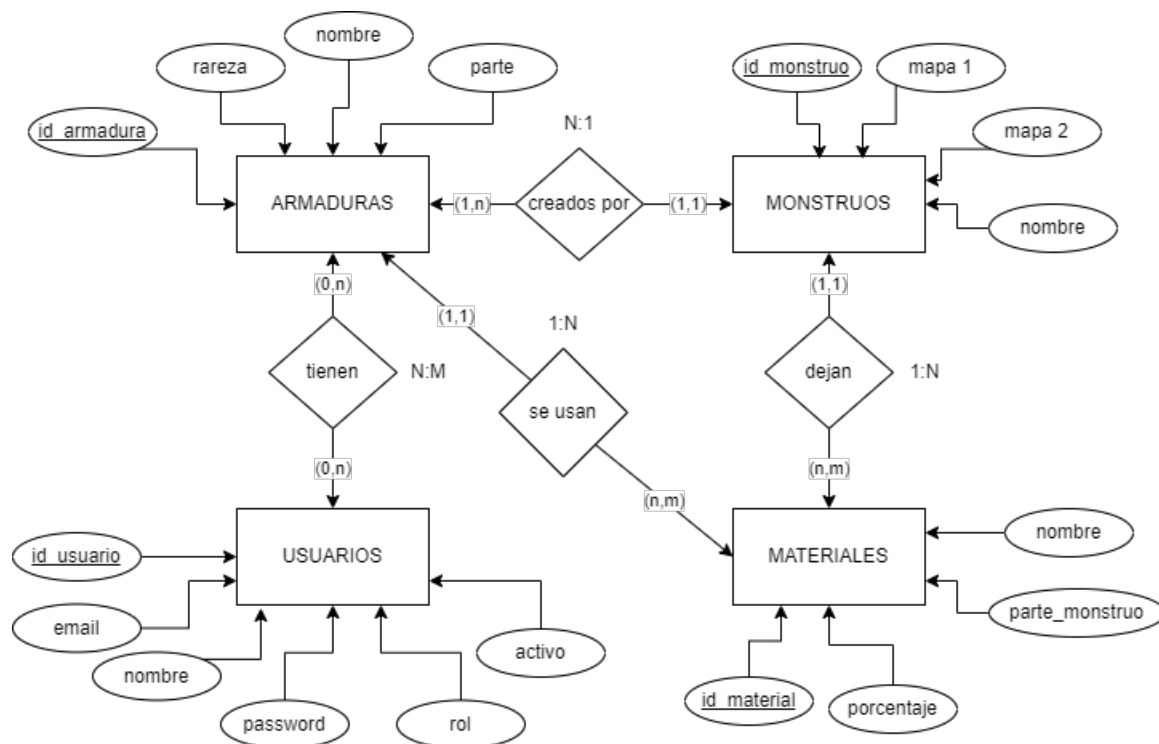
HTML: Lenguaje de marcas de hipertexto, componente básico de una web. Utilizado para la creación y la presentación de la vista de la aplicación. Junto con las hojas de estilo css formamos toda la presentación de nuestra página al usuario.

CSS



CSS: Lenguaje de marcas utilizado para dar estilos a nuestros documentos html, controlando como luce nuestra página web. Ayuda a potenciar el alcance de nuestro contenido frente a la aplicación de estilos directamente al documento html.

Modelo de Datos



- **Usuarios:** Tabla que almacenará los datos de los usuarios y administradores registrados en la aplicación.
 - id_usuario (int): Código identificador para cada usuario.
 - nombre (varchar): Alias del usuario.
 - contraseña (varchar): Contraseña del usuario.
 - correo (varchar): Correo electrónico del usuario.
 - rol (varchar): Rol en la aplicación del usuario (usuario normal o administrador).
- **Armaduras:** Tabla que almacenará los datos de las armaduras del videojuego.
 - id_armadura (int): Código identificador para cada armadura.
 - nombre_armadura (varchar): Nombre de la armadura.
 - rareza (varchar): Clasificación enumerada de la rareza de la armadura.
 - parte (varchar): Pieza de la armadura completa (casco, guantes, botas...).



MHRise Optimizer

Luis Rafael Arroyo Caballero

- **Monstruos:** Tabla que almacenará los datos de los monstruos del videojuego.
- id_monstruo (int): Código identificador para cada monstruo.
- nombre_monstruo (varchar): Nombre del monstruo.
- mapa_1 (varchar): Primer mapa en el que se puede encontrar al monstruo.
- mapa_2 (varchar): Segundo mapa en el que se puede encontrar al monstruo.
- **Materiales:** Tabla que almacenará los datos de los materiales del videojuego.
- id_material (int): Código identificador para cada material.
- nombre_material (varchar): Nombre del material.
- porcentaje (varchar): Porcentaje que tenemos para obtener el material en cuestión.
- parte_monstruo (varchar): Parte del monstruo que a romper para obtener poder obtener el material.
- **Inventario:** Tabla que surge de la relación usuarios-armaduras, almacenará el equipamiento guardado por el usuario.
- cantidad_material (int): Cantidad deseada por el usuario del material.



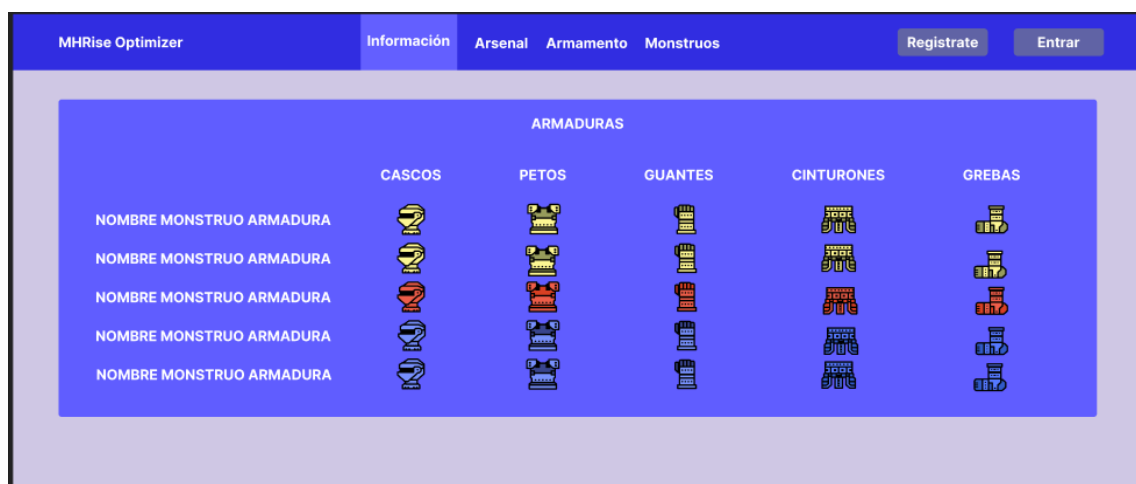
Prototipo de la Aplicación Web



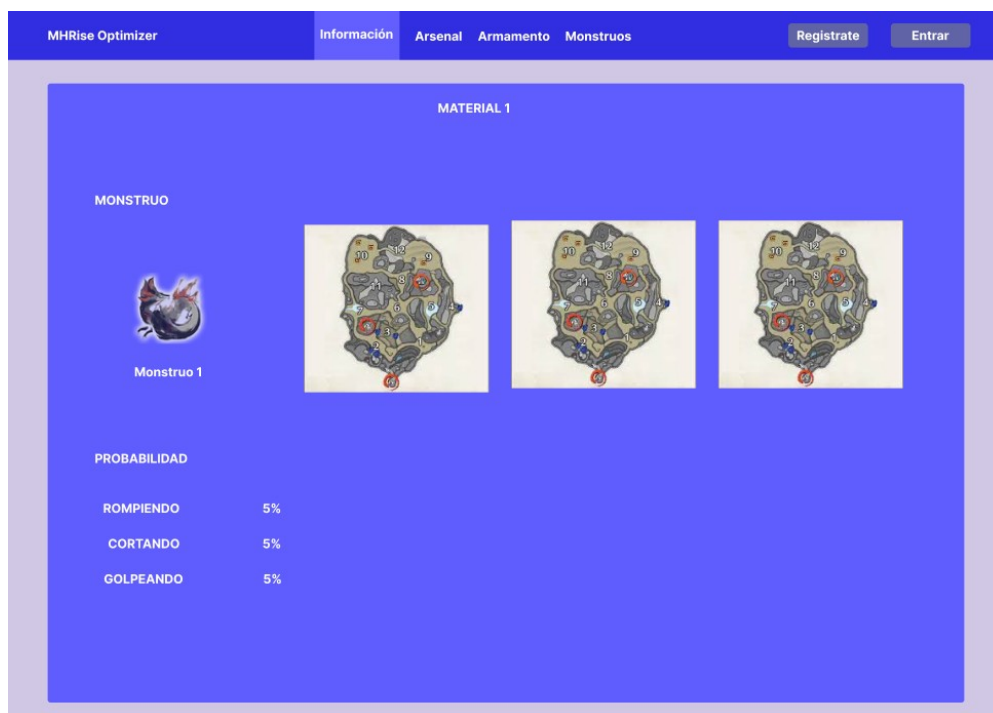
Página de inicio la cual contiene información sobre el videojuego y la página web. Además de contener enlaces de interés a páginas relacionadas con la creación de personaje y páginas de los desarrolladores con información de toda la franquicia.



Pantalla que contiene las armaduras almacenadas por los usuarios. No podrá ser accesible sin iniciar sesión y los usuarios pueden tanto añadir como eliminar armaduras de su arsenal.



Pantalla que muestra todas las armaduras, accesible tanto para usuarios que hayan iniciado sesión como para los que no lo hayan hecho. Sin embargo los usuarios que hayan iniciado sesión tendrán disponible la funcionalidad de añadir armaduras a su arsenal.



Pantalla que muestra los datos de los monstruos del juego. Muestra información como en la zona donde podemos encontrar estos enemigos como la probabilidad de conseguir objetos.



Definición API REST

Pantalla de inicio ("/ "): Página principal del proyecto. Contiene información general relacionada con el juego, la comunidad y el motivo de la aplicación.

Pantalla de registro ("/register"): Formulario de registro, donde se deberán rellenar los campos de usuario nombre, contraseña y correo electrónico.

Pantalla de inicio de sesión ("/login"): Formulario para iniciar sesión. Se deberán rellenar los campos nombre y contraseña para verificar que el usuario ya esta registrado en la base de datos.

Pantalla de cierre de sesión ("/logout"): Función para cerrar la sesión iniciada en ese momento.

Usuarios ("/usuarios"): Página disponible para administradores, que nos muestra todos los usuarios registrados en la base de datos.

Borrar usuario ("/usuarios/removeUsuario") [param Código(String)]: Función para borrar usuarios solo disponible para administradores. Recibira un código (id_usuario) para borrar el usuario correspondiente de la base de datos.

Crear usuario ("/usuarios/createUser"): Formulario para crear un usuario similar al registro. Solo disponible para administradores.

Actualizar usuario ("/usuarios/updateUser") [param Código(String)]: Formulario para modificar un usuario existente en la base de datos. Solo disponible para administradores. Recibirá un código (id_usuario) para modificar el usuario correspondiente de la base de datos.



Pantalla de armaduras ("/armaduras"): Pantalla que nos mostrará las armaduras registradas en la base de datos.

Pantalla de monstruos ("/monstruos"): Pantalla que nos mostrará el listado de monstruos registrados en la base de datos y su información.

Pantalla de arsenal ("/arsenal"): Pantalla que nos mostrará el listado de armaduras vinculadas por un usuario.

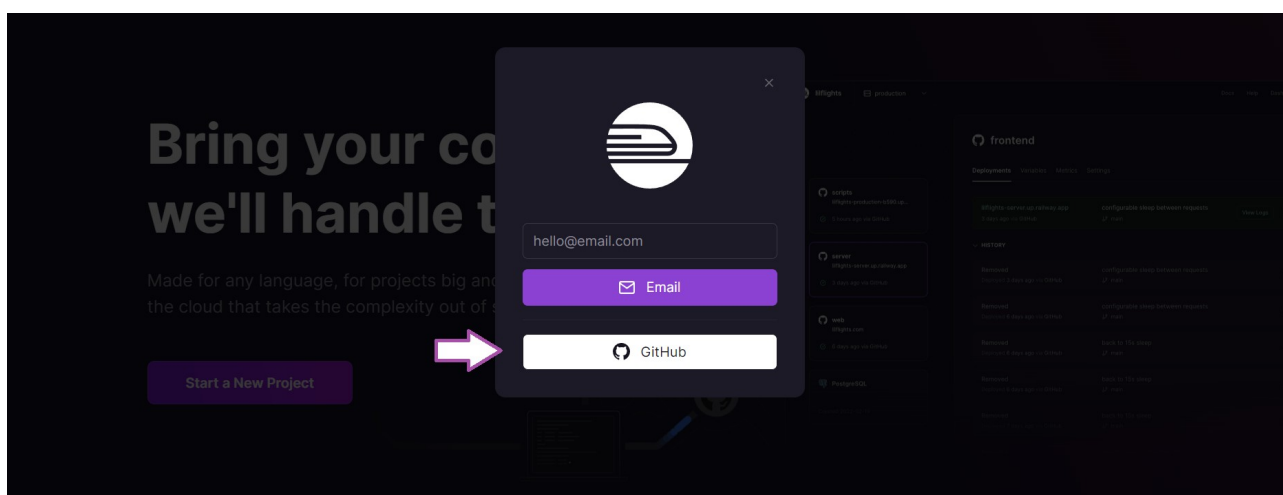
Añadir armadura (/arsenal/add) [param Código(String)]: Función que permitirá al usuario añadir una armadura a su arsenal.



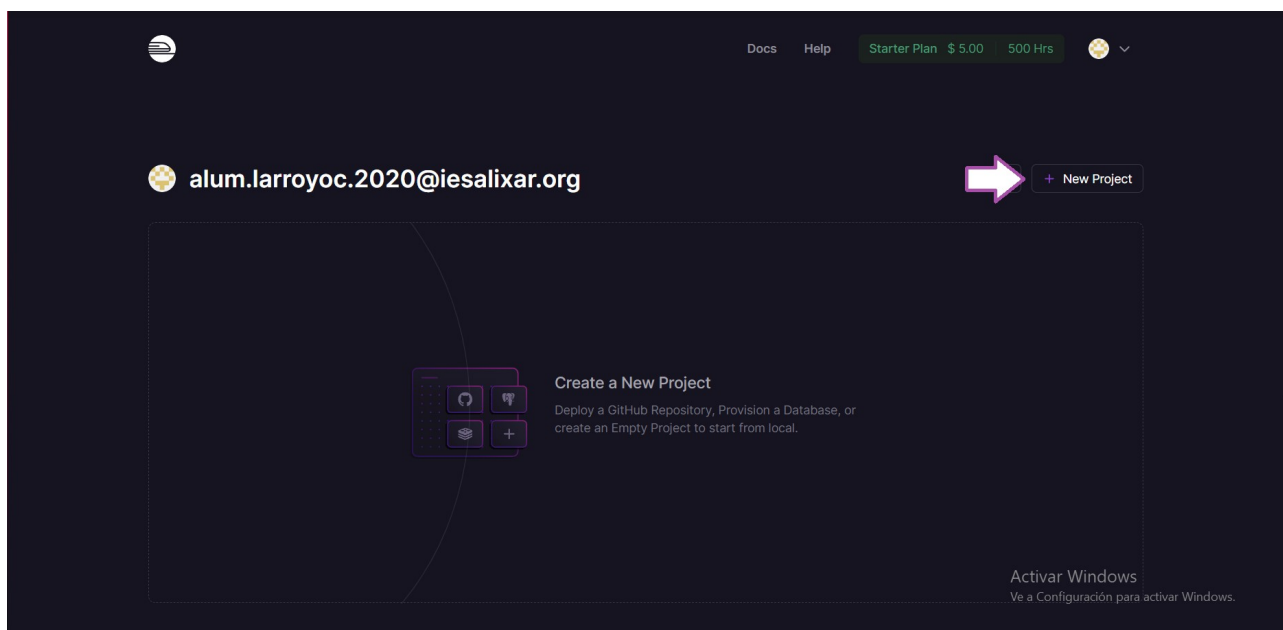
Manual de Despliegue

El despliegue de nuestra aplicación se ha realizado mediante la herramienta Railway, una aplicación web que nos permite vincular nuestro repositorio de GitHub para desplegar de manera muy sencilla el proyecto. Para desplegar el proyecto hay que seguir los siguientes pasos.

- Iniciar sesión con nuestra cuenta de GitHub, donde tenemos alojado nuestro repositorio con el proyecto a desplegar.

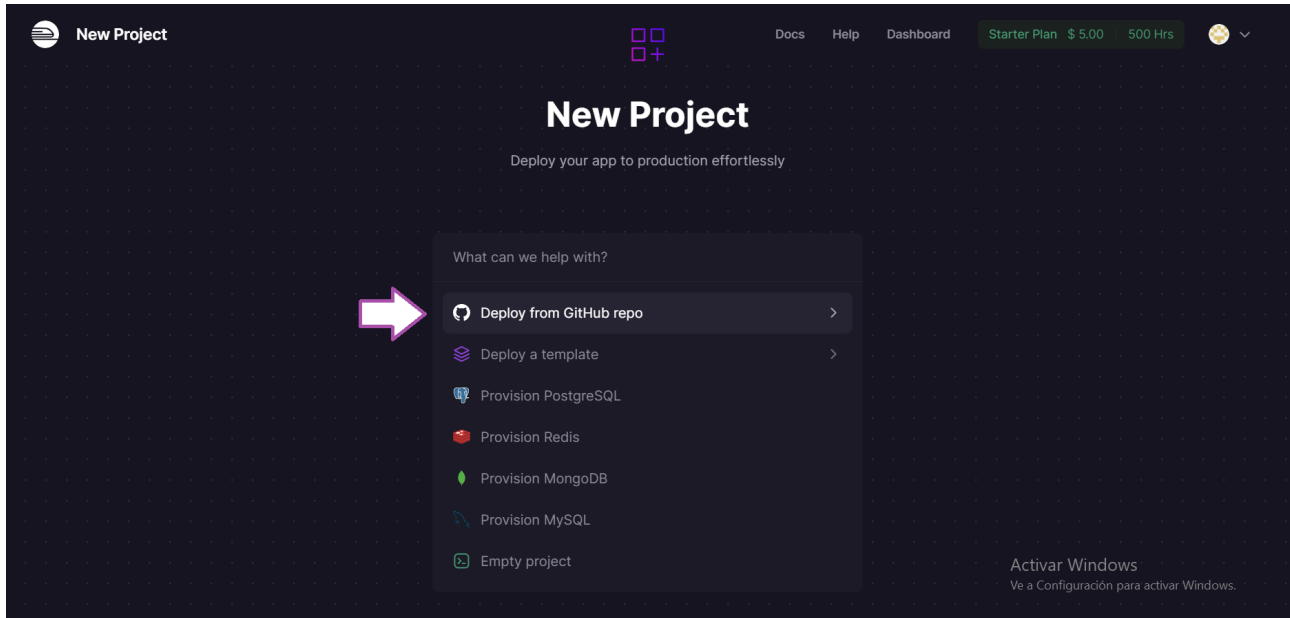


- Una vez iniciada la sesión, haremos click en nuevo proyecto (**New Project**).

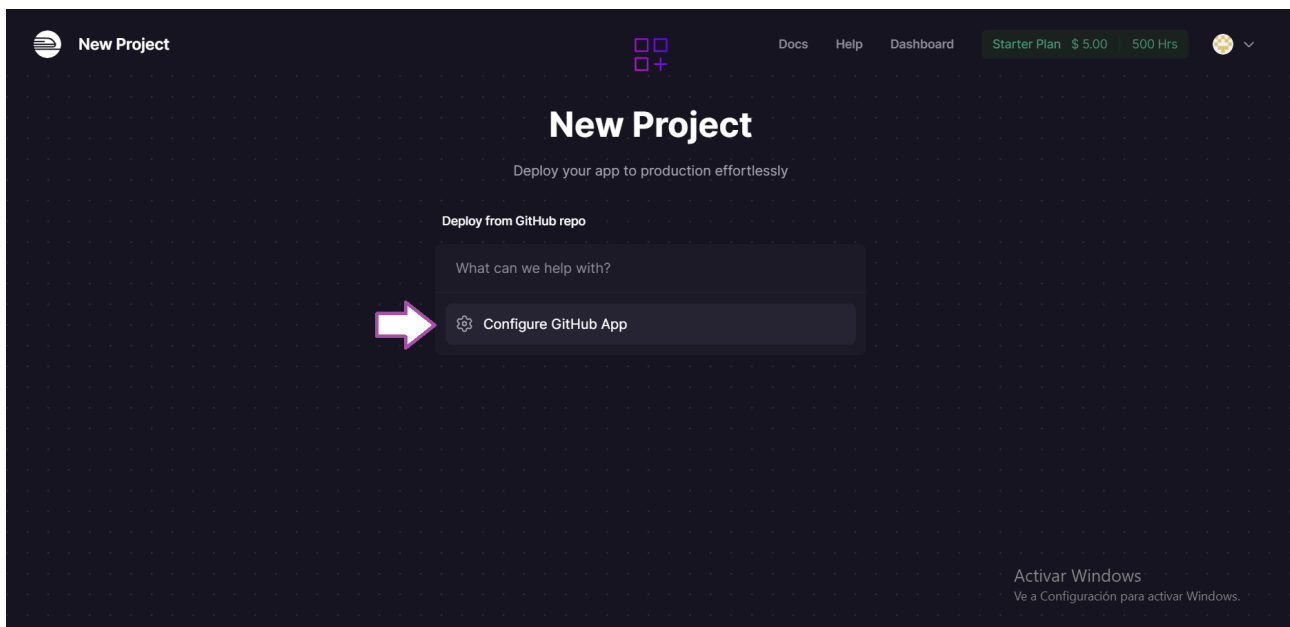




- Una vez nos encontremos en esta pantalla, seleccionaremos la opción desplegar desde repositorio de GitHub (**Deploy from Github repo**).

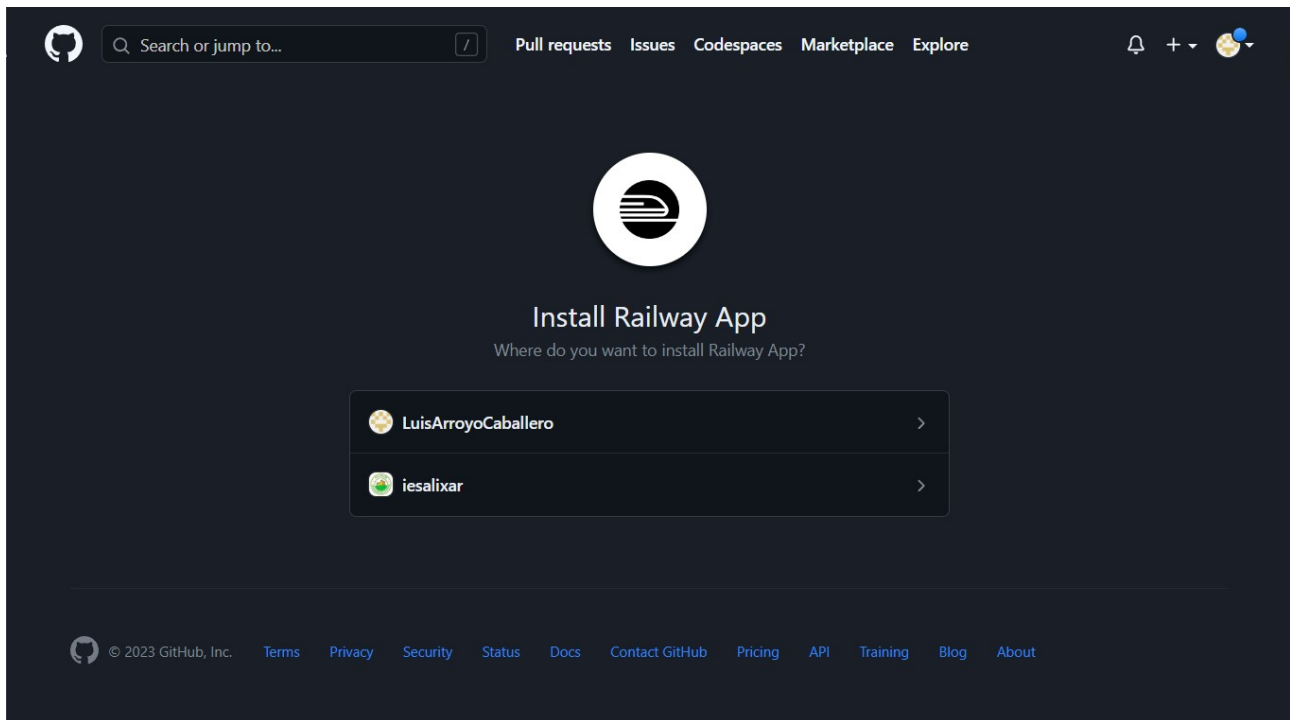


- Seleccionamos la opción configurar la aplicación de GitHub (**Configure GitHubApp**).

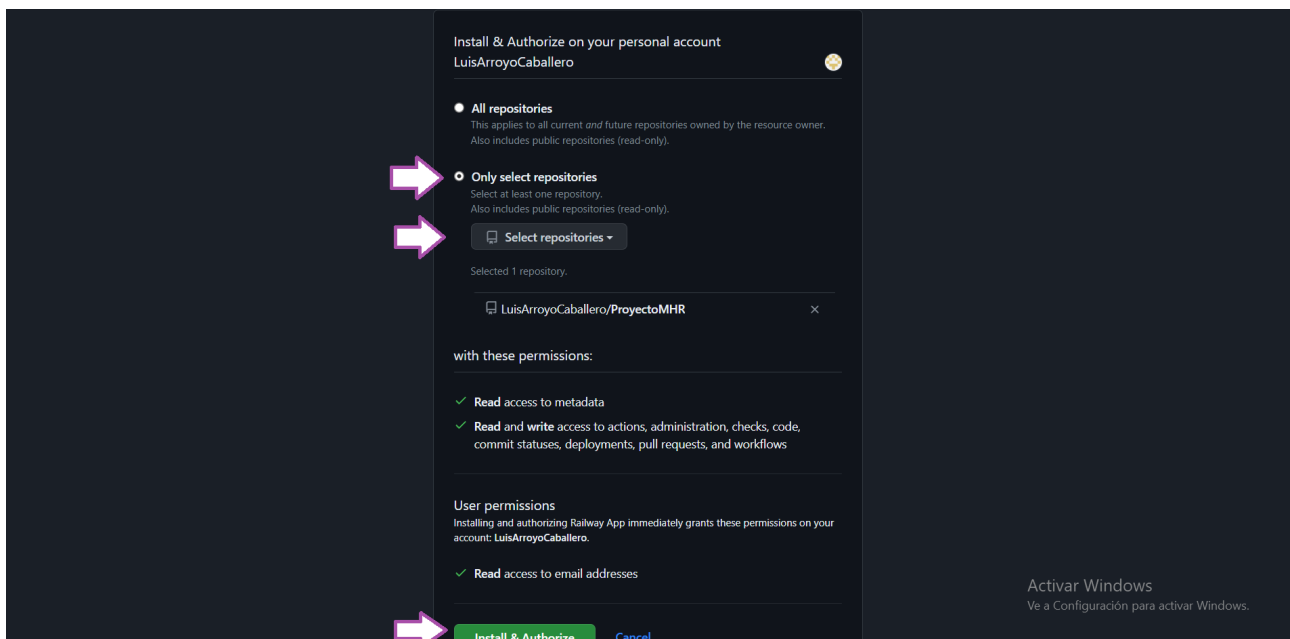




- Y seleccionamos nuestro usuario en el que instalaremos Railway App.

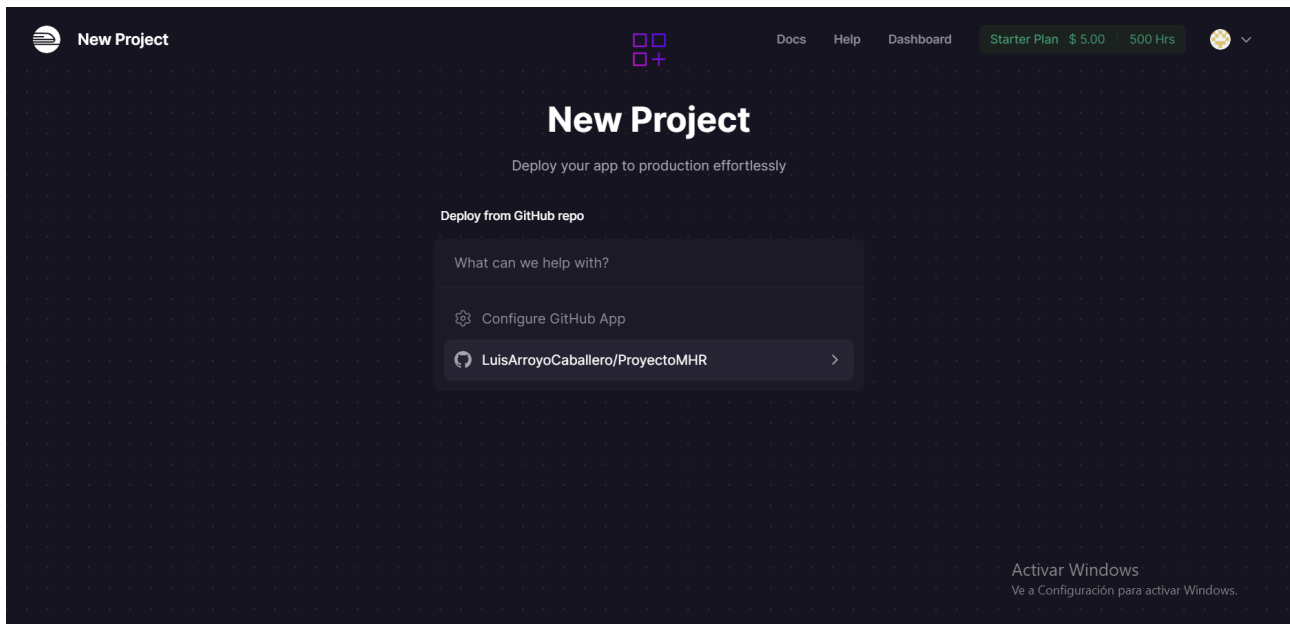


- Se abrirá una ventana emergente, en la que indicaremos el repositorio en cuestión para desplegar la aplicación.

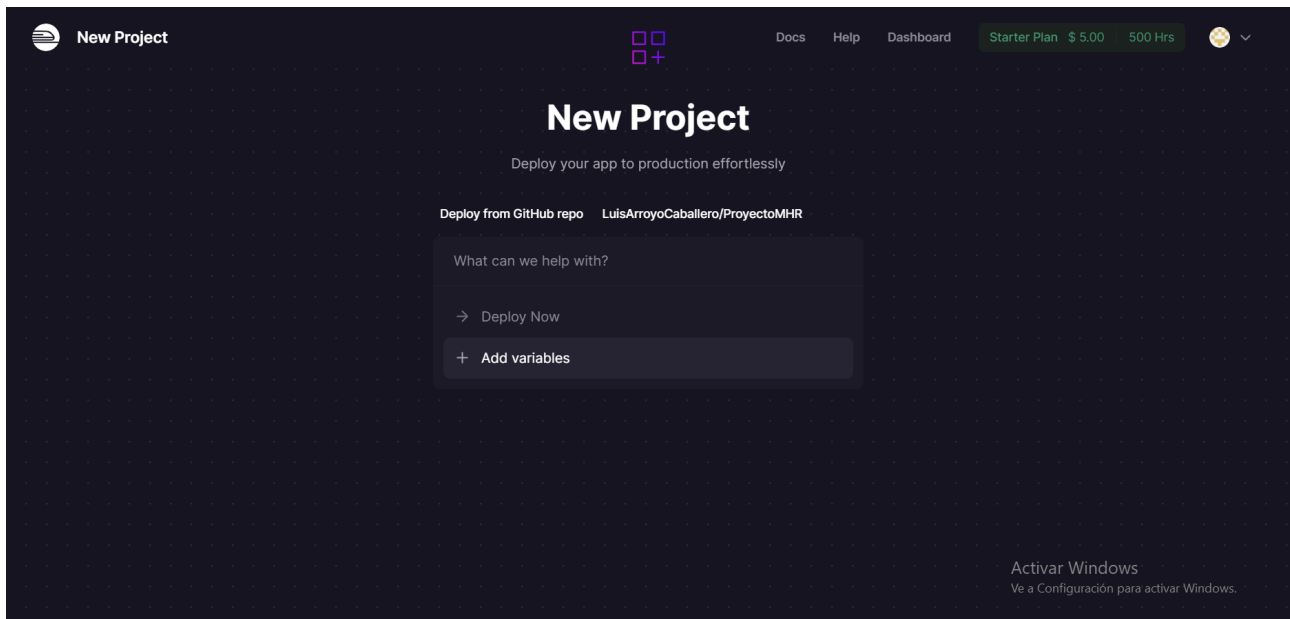




- Seleccionamos el proyecto.

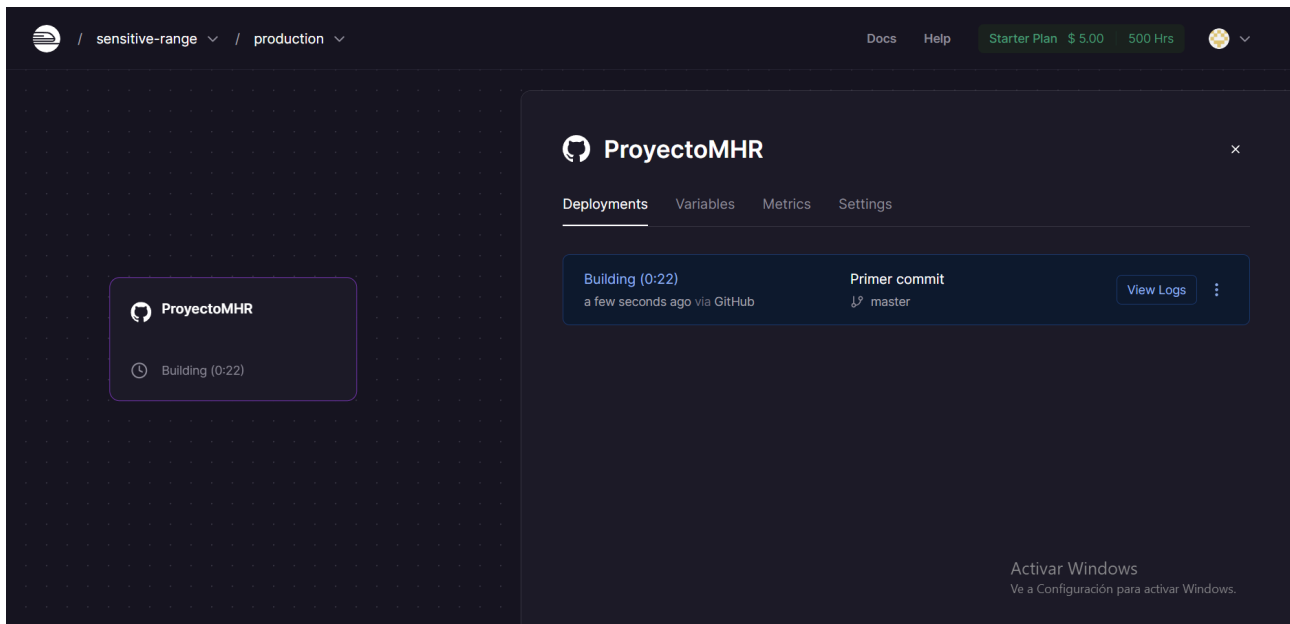


- Seleccionamos la opción desplegar ahora (**Deploy now**).

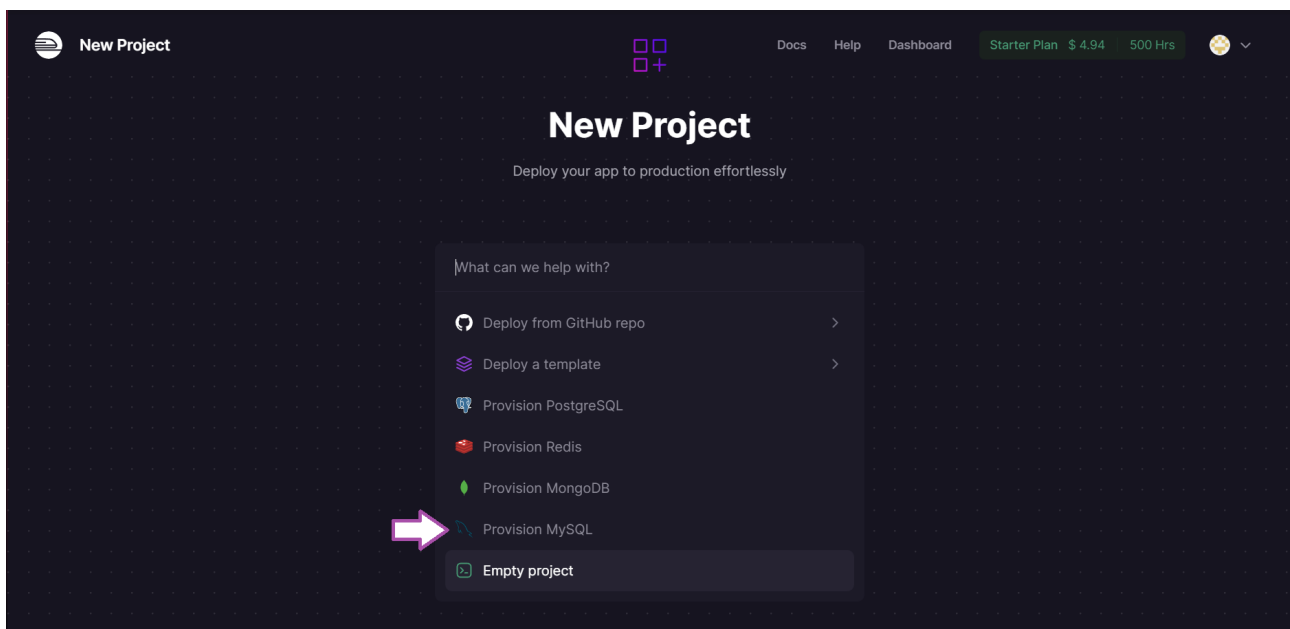




- Y ya tendríamos en proceso un primer despliegue nuestra aplicación.

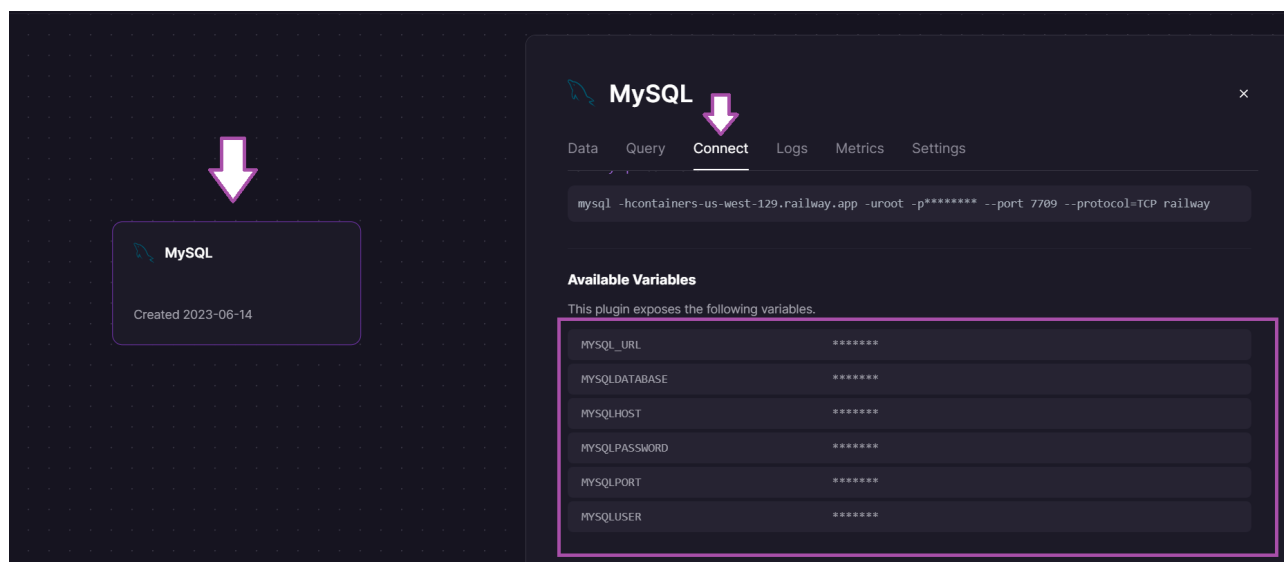


- Ahora montaremos la base de datos, también en Railway. Para ello crearemos un nuevo proyecto, pero esta vez será una base de datos mySQL. Haremos click en **Provison mySQL**.

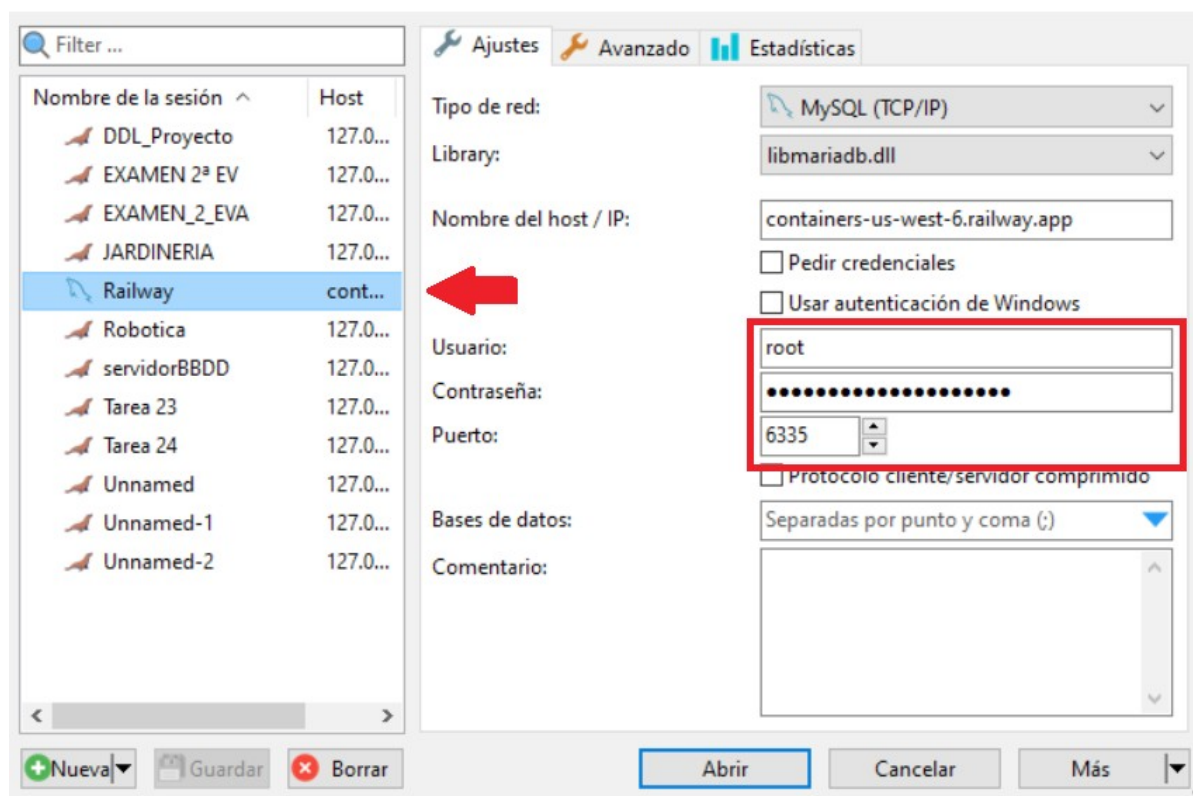




- Una vez seleccionado el tipo de base de datos a crear, haremos click en el recuadro de la base de datos, nos iremos a las opciones de conexión (**Connect**) y buscaremos el apartado de variables para incluir en nuestro fichero de propiedades del proyecto.



- Ya localizadas las variables de la base de datos en railway, vamos a importar la base de datos. Realizaremos una nueva conexión con nuestro gestor de base de datos (en mi caso HeidiSQL) utilizando las variables de la base de datos de Railway.





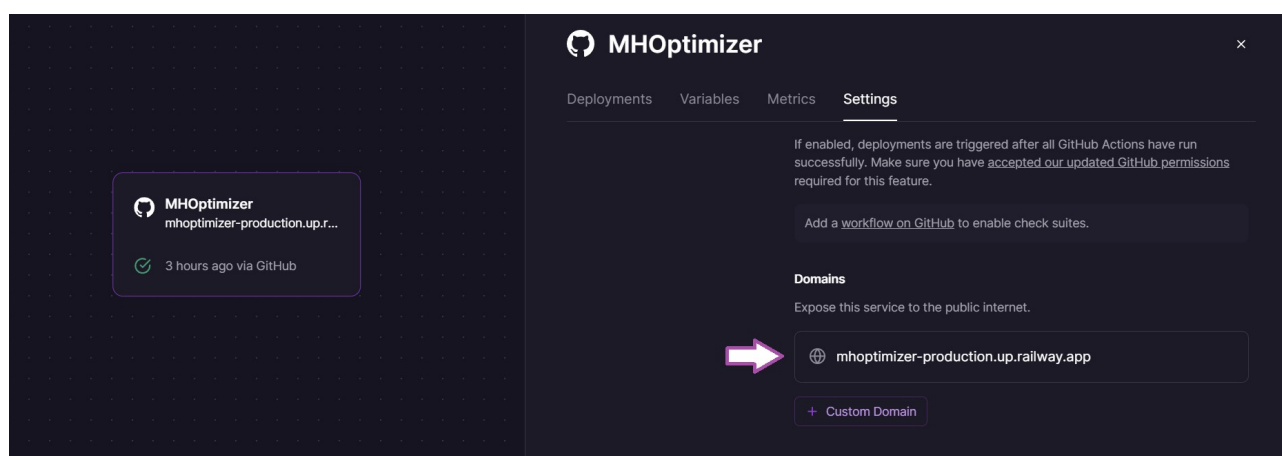
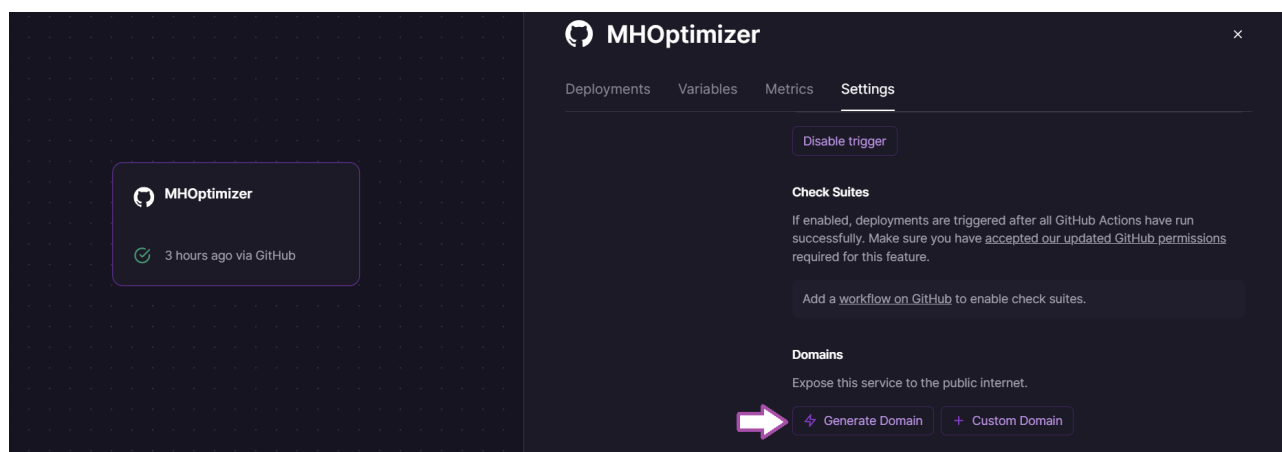
MHRise Optimizer

Luis Rafael Arroyo Caballero

- Utilizaremos nuestro script para migrar la base de datos a la que esta alojada en Railway, este proceso puede variar dependiendo del gestor de base de datos. Y con esto tendríamos lista la base de datos, y solo nos faltaría incluir las variables en nuestro application.properties.

```
application.properties
1 spring.datasource.url=jdbc:mysql://root:Niluo83ioQp9vJu8DX5J@containers-us-west-6.railway.app:6335/railway
2 spring.datasource.username=root
3 spring.datasource.password=Niluo83ioQp9vJu8DX5J
4 spring.jpa.hibernate.ddl-auto=update
5 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
6 spring.jpa.show-sql = true
7 spring.datasource.driver-class-name= com.mysql.jdbc.Driver
8 server.port = 8383
```

- Ya tendríamos lista nuestra página web, en caso de necesitar actualizar el proyecto, al reflejarlo en nuestro repositorio Railway se encargará de volver a desplegar el proyecto. Para generar el enlace de nuestra página, nos iremos a nuestros proyectos de Railway e iremos a las opciones del proyecto. En el apartado **Enviroment** nos aparecerá una opción para generar un enlace. Al seleccionarla nos generará automáticamente el link a nuestro proyecto ya desplegado.





Postmortem y conclusiones del proyecto

El proceso de ejecución de este proyecto fue desde un principio bastante complejo. Al tratar sobre un videojuego la gestión y creación de la base de datos ha sido bastante difícil de abordar.

Por esta razón la página tiene algunos puntos que mejorar, entre ellos, el estilo y apariencia de la de la misma teniendo una interfaz simple y anticuada, la cual debería de modificarse de manera prioritaria y unos estilos poco atractivos para el usuario.

Por otro lado deberían de añadirse varias funcionalidades, como buscadores para encontrar de una manera mas sencilla tanto el monstruo, armadura u objeto en cuestión.

Por último, algunas de las mecánicas del videojuego no se han visto reflejadas en la página web, debido a que hay constantes actualizaciones del mismo, siendo bastante complejo añadir durante el desarrollo nuevos tipos de objeto, diferentes mejoras para las armaduras ya existentes o nuevas versiones de los monstruos, por lo que sería una mejora a tener en cuenta.

Para concluir, la idea del proyecto tiene mucho potencial y definitivamente vería uso en la comunidad de este juego, sin embargo deberían realizarse ciertos cambios y mejoras para que sea más sencilla de usar y atractiva para los usuarios.