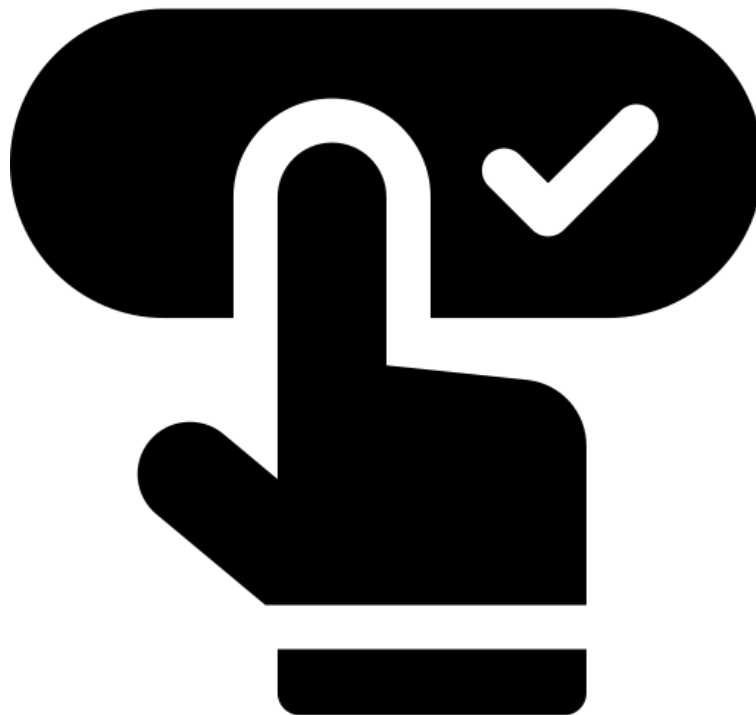


RESERVA LA PISTA

<https://reservalapista.up.railway.app/>



Proyecto Final – Grado Superior

2º Desarrollo de Aplicaciones Web – 2022/23

Alumno: Sebastián Bermúdez Herrera

Tutor: D. Antonio Gabriel González Casado

I.E.S. Alixar – Castilleja de la Cuesta (Sevilla)

Fechas: 15/09/2022 – 11/09/2022

ÍNDICE:

1. [Descripción del proyecto.](#)
2. [Definir el objetivo de la aplicación.](#)
3. [Stack tecnológico.](#)
4. [Alcance.](#)
5. [Esquema de la Base de Datos.](#)
6. [Manual de usuario.](#)
7. [Manual de administrador.](#)
8. [Manual de despliegue.](#)
9. [Conclusión](#)

1.- Descripción del proyecto.

ReservaLaPista es una aplicación web donde podrás reservar pistas deportivas.

Normalmente, cuando quieres reservar una pista, por ejemplo, en tu polideportivo municipal, debes de llamar o presentarte en la oficina. Esta aplicación está pensada para que desde tu casa y sin tener que contactar con nadie lo puedas hacer en un simple clic, sin depender de nadie, solamente con internet.

Y es que, cada vez, el uso de estas tecnologías es más frecuentes y son de gran ayuda para las personas.

Cuenta con varias funciones, pero la principal es la de reservar espacios.

¿Por qué no te atreves y la usas?

2.- Definir el objetivo de la aplicación.

El propósito de esta aplicación es poder facilitarles a las personas el poder alquilar pistas deportivas, sin tener que hacer ninguna gestión extra.

Mediante esta aplicación conectamos a los usuarios directamente a nuestra base de datos, en la que ellos mismos pueden ver la disponibilidad horaria de cada espacio deportivo.

Es una web muy intuitiva y clara, fácil de usar para todas las edades, sin necesidad de tener conocimiento avanzados de informática.

A parte de ayudar a los usuarios a la gestión de reservas, también es de gran ayuda para la empresa puesto que no tendría que depender de una persona encargada de realizar las reservas manualmente.

Hoy en día, la mayoría de los jóvenes que práctica deporte en pistas deportivas realizan las reservas desde alguna aplicación web. Como por ejemplo 'Playtomic', una de las aplicaciones más usada y popular, sobre todo para los amantes del pádel y tenis. Esta aplicación ofrece a las empresas un espacio en su web para mostrar sus pistas deportivas y que los usuarios puedan reservar. Una ventaja de 'Playtomic' es que aparte de poder reservar pistas con un simple clic, puedes reservar clases particulares. Además, también tiene un apartado para buscar partidos con otros usuarios de la plataforma. Otra aplicación relacionada es 'Booksy', en la cual podrás encontrar cualquier tipo de servicios a tu alrededor, como por ejemplo peluqueros, entrenadores personales, salón de tatuajes, etc.

Con esto quiero dejar claro que este tipo de aplicaciones las usamos en el día a día, y hasta la propia Universidad de Sevilla tiene una web para la reserva de aulas.

Por lo tanto, si tienes una empresa de este tipo, que ofrezca servicios a las personas, creo que debería de estar dado de alta en alguna de estas aplicaciones.

Los requisitos funcionales de mi aplicación son:

REQ01-Login: El usuario deberá loguearse con su cuenta para poder entrar. Si no tiene cuenta podrá crearse una en el formulario de crear cuenta.

REQ02-Alta de usuario: El usuario sin cuenta registrada podrá crearse una cuenta nueva.

REQ03-Baja de usuario: El usuario puede darse de baja cuando desee y su cuenta será eliminada.

REQ04-Reserva de espacios: El usuario puede ver la disponibilidad de las pistas y realizar reservas.

REQ05-Mis reservas: El sistema deberá mostrar al usuario las reservas que ha realizado. El usuario podrá cancelar su reserva siempre y cuando no haya empezado el partido (hora de inicio).

REQ06-Todas las reservas: El sistema mostrará todas las reservas realizadas. Solo tendrá acceso los administradores. Podrán realizar gestiones como eliminar o crear nueva reserva y ver información del usuario de la reserva.

REQ07-Ajustes de usuario: El sistema cuenta con un apartado de ajustes para que el usuario modifique sus datos.

REQ08-Logout: El sistema cuenta con un botón logout en la barra superior, para que el usuario puede cerrar sesión cuando quiera.

REQ09-Eliminar reserva: El sistema tiene un apartado para cancelar la reserva.

REQ10-Horarios: En este apartado el usuario puede ver las pistas y sus horarios.

REQ11-Pistas: El sistema mostrará todas las pistas. Solo los usuarios administradores tienen permiso para gestionarlas (añadir, editar, ver reservas de la pista o eliminar pista)

REQ12-Usuarios: En este apartado se mostrarán todos los usuarios. Solo los administradores tendrán acceso y podrán realizar las siguientes gestiones: añadir, editar, ver información, ver reservas y eliminar usuario.

3.- Stack tecnológico.

Para la parte del back he decidido utilizar **Spring Boot** ya que es uno de los framework libre más usados de JAVA. Además, tiene la posibilidad de incorporar **Spring Security**, el cual también he usado para la autenticación y control de acceso.

En cuanto a la parte del front usaré **Thymeleaf**, un moderno motor de plantillas Java del lado del servidor para entornos web. También utilizaré **HTML** y **CSS**. Además, implementaré **Bootstrap** para usar su gran variedad de componentes y hacer que mi web sea responsive. Con **Javascript** haré que la aplicación sea más interactiva.

Para la creación de las plantillas HTML me he ayudado de **Visual Studio Code** ya que es un muy buen editor de código fuente, libre y con gran variedad de plugins.

El software que utilizaré es **Spring**, bajo un servidor local **Tomcat**. Algunas de las dependencias de Spring que implemento son:

- **Spring Boot:** Facilita la creación de aplicaciones Spring y contiene servidor local Tomcat. Puede crear filtro para diferenciar a que página puede ir el usuario dependiendo por ejemplo del rol que tenga.
- **Spring Security:** es un marco de autenticación y control de acceso potente y altamente personalizable.
- **Thymeleaf:** un moderno motor de plantillas Java del lado del servidor para entornos web.
- **Spring DATA JPA y MariaDB JDBC (local):** para controlar los datos de almacenamiento SQL usando Hibernate.
- **Tomcat:** es un contenedor de servlets que se puede usar para compilar y ejecutar aplicaciones web realizadas en Java. Implementa y da soporte tanto a servlets como a páginas JSP.
- **Spring Boot Mail:** para enviar correos electrónicos.
- **MYSQL:** para controlar los datos de almacenamiento SQL. Este solo lo uso en el servidor web donde está alojado mi web.
- Para la gestión de la base de datos usare:
 - EN LOCAL:
 - **MariaDB** en un contenedor **Docker**. Me ayudaré de **DBeaver**, que es una interfaz para la gestión de bases de datos.
 - EN SERVIDOR RAILWAY:
 - **SQL**.

4.- Alcance.

A continuación, mostraré las tareas que mi proyecto llevará a cabo:

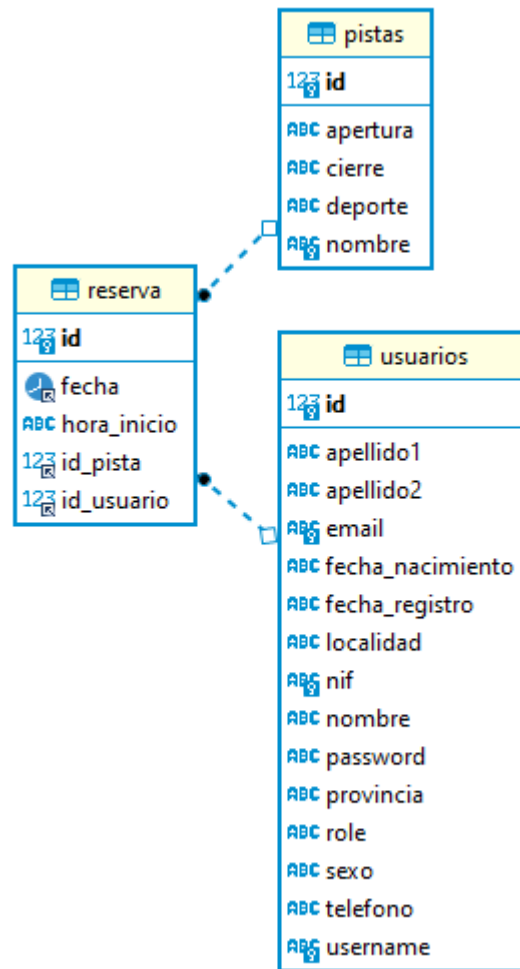
Para poder acceder a la web el usuario debe de estar registrado.

Hay dos tipos de roles de usuarios:

- *Usuario (ROL_USER):*
 - Puede acceder a la web.
 - Puede realizar una reserva.
 - Si la reserva aún no ha comenzado, puede cancelar la reserva.
 - Puede ver las pistas disponibles y sus horarios.
 - Puede editar sus datos de usuario.
 - Puede ver su historial de reservas.
 - Puede dar de baja su cuenta.

- *Usuario administrador (ROL_ADMIN):*
 - Puede acceder a la web.
 - Gestiona todos los usuarios registrados: añadir, eliminar, ver información de usuario y sus reservas.
 - Puede realizar reservas para cualquier usuario.
 - Gestiona todas las pistas: crear, editar, eliminar y ver sus reservas.
 - Tiene el control de las reservas realizadas. Puede eliminar cualquier reserva.
 - Puede modificar tanto sus datos como los de cualquier usuario.

5.- Esquema de la Base de Datos.



Código fuente – Script SQL

```

-- reservas_de_pistas.pistas definition
CREATE TABLE `pistas` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `apertura` varchar(255) NOT NULL,
  `cierre` varchar(255) NOT NULL,
  `deporte` varchar(255) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_1730s9pd0bl6nw5e6v45546np` (`nombre`)
) ENGINE=InnoDB AUTO_INCREMENT=42 DEFAULT CHARSET=latin1;
  
```



```

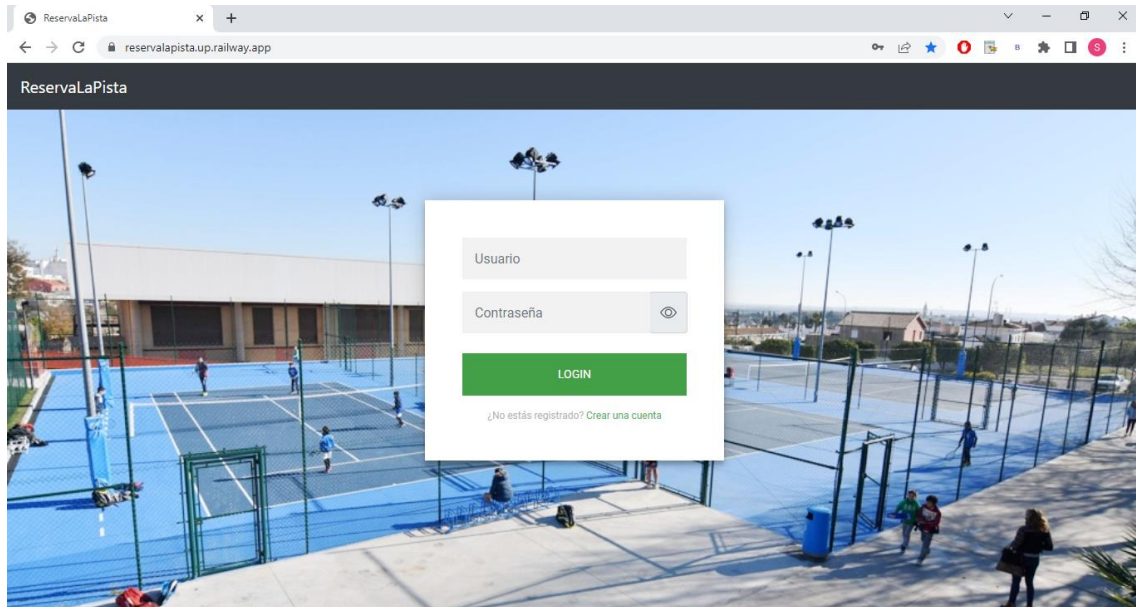
-- reservas_de_pistas.usuarios definition
CREATE TABLE `usuarios` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `apellido1` varchar(255) NOT NULL,
  `apellido2` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `fecha_nacimiento` varchar(255) NOT NULL,
  `fecha_registro` varchar(255) NOT NULL,
  `localidad` varchar(255) NOT NULL,
  `nif` varchar(9) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `provincia` varchar(255) NOT NULL,
  `role` varchar(255) NOT NULL,
  `sexo` varchar(255) NOT NULL,
  `telefono` varchar(9) NOT NULL,
  `username` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_kfsp0s1tflm1cwlj8idhqsad0` (`email`),
  UNIQUE KEY `UK_igfhcb3km5lfdcc9v0v3mdbbg` (`nif`),
  UNIQUE KEY `UK_m2dvbwfge291euvmk6vkkocao` (`username`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1;

-- reservas_de_pistas.reserva definition
CREATE TABLE `reserva` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `fecha` date DEFAULT NULL,
  `hora_inicio` varchar(255) DEFAULT NULL,
  `id_pista` bigint(20) DEFAULT NULL,
  `id_usuario` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UKhkfs93i70x726y4ihs7iu7pyn` (`fecha`,`hora_inicio`),
  KEY `FK1ifa470ho0dqxsffku0ka0c2` (`id_pista`),
  KEY `FKj8cxeodpmehovje6152y4ywqm` (`id_usuario`),
  CONSTRAINT `FK1ifa470ho0dqxsffku0ka0c2` FOREIGN KEY (`id_pista`) REFERENCES `pistas` (`id`),
  CONSTRAINT `FKj8cxeodpmehovje6152y4ywqm` FOREIGN KEY (`id_usuario`) REFERENCES `usuarios` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=111 DEFAULT CHARSET=latin1;

```

6.- Manual de usuario.

Login. La primera pantalla con la que nos encontramos es el Login. Nos muestra un formulario de autenticación en el cual debemos introducir nuestro usuario y contraseña. Para crear una cuenta podemos hacer click en el enlace “Crear una cuenta”.



Registro. Nos muestra un formulario que debemos rellenar con nuestros datos.

Usuario, DNI y Email son claves únicas, por lo tanto, no puede haber en la base de datos dos registros iguales. En el caso de querer registrar 2 cuentas con algunos de estos campos duplicados se nos mostrará un error.

Una vez hayamos creado nuestra cuenta, se nos redireccionará al Login donde debemos autenticarnos introduciendo usuario y contraseña.

ReservaLaPista

reservalapista.up.railway.app/register

Usuario

Contraseña

Email

Nombre

Primer Apellido

Segundo Apellido

DNI

Fecha de nacimiento

dd/mm/aaaa

Sexo

Telefono

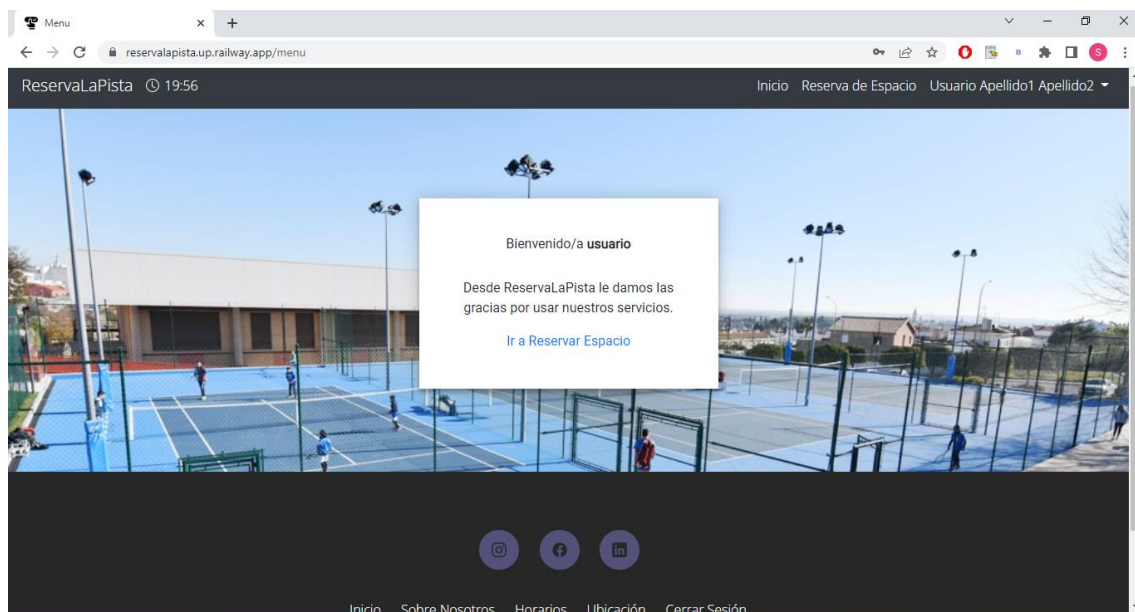
Localidad

Provincia

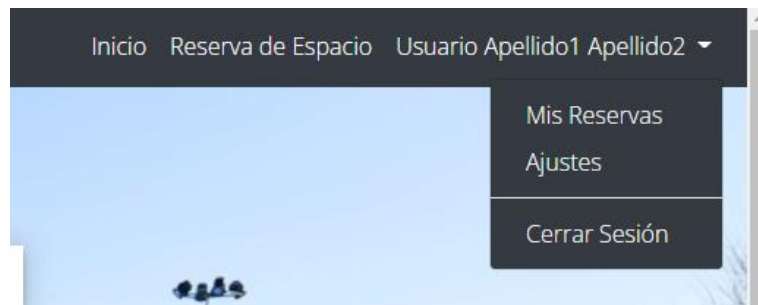
REGISTRAR

¿Estás registrado? Iniciar Sesión

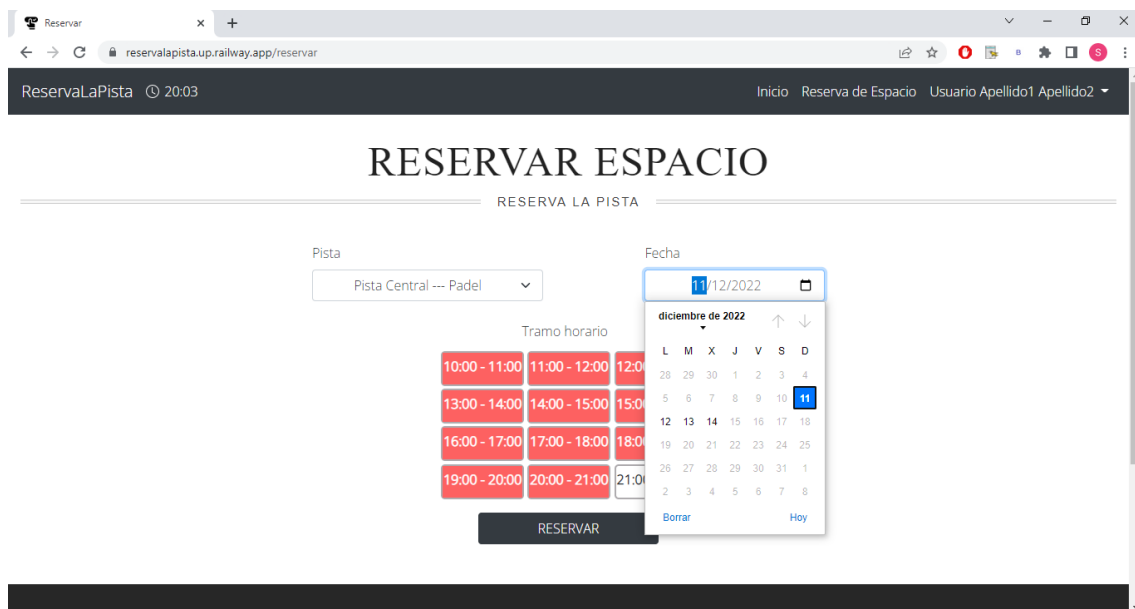
Inicio. Una vez que ya nos hemos identificado podemos ver esta pantalla. En ella nos muestra un mensaje de bienvenida con un enlace para reservar.



En el header podemos apreciar que cuenta con varios enlaces y un menú desplegable con más opciones. Una de ellas la de “Cerrar sesión”.



Reservar espacio. En esta pantalla es donde el usuario podrá realizar las reservas. Para ello deberá seleccionar una pista de la lista seleccionable, elegir una fecha y un tramo horario.



En cuanto a las fecha, solo podrá reservar desde el día actual hasta dentro de 3.

```

1 //--- BLOQUEAR FECHAS VENCINAS CALENDARIO DE FECHAS ---
2 // Obtener fecha actual
3 let fecha = new Date();
4 // Obtener cadena en formato yyyy-mm-dd, eliminando zona y hora
5 let fechaMin = fecha.toISOString().split('T')[0];
6
7 // Agregar 3 días
8 fecha.setDate(fecha.getDate() + 3);
9 // Obtener cadena en formato yyyy-mm-dd, eliminando zona y hora
10 let fechaMax = fecha.toISOString().split('T')[0];
11
12 // Asignar valor mínimo
13 document.querySelector('#fecha').min = fechaMin;
14 // Asignar valor máximo
15 document.querySelector('#fecha').max = fechaMax;
16

```

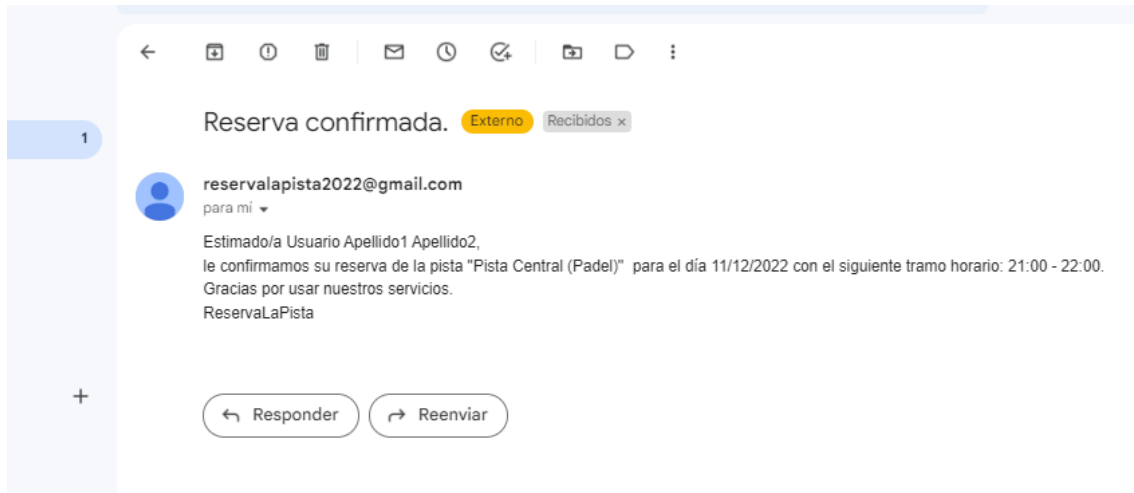
Y en los tramos horarios, a medida que vaya avanzado el día y la hora actual supere el tramo, se irá deshabilitando y poniendo en rojo las horas ya pasadas.

```

53 function getFechaInput() {
54   // cogemos el elemento del html
55   var fechaElement = document.getElementsByName("fecha");
56   // Obtenemos la fecha actual
57   let dia = new Date()
58   // extraemos cadena de dia, para quedarnos solo con la fecha (yyyy-mm-dd)
59   let diaActual = dia.toISOString().substring(0, 10)
60
61   //desde 0 hasta el tamaño del elemento de fechaElement
62   for (var i = 0; i < fechaElement.length; i++) {
63     // va recorriendo cada valor de fecha y saca valor a valor
64     const fecha = fechaElement[i].value
65     //si el dia actual es igual a la fecha
66     if (diaActual === fecha) {
67       //CAMBIAR COLOR A LOS LABEL QUE YA HA PASADO LA HORA
68       for (var i = 0; i < labelBoton.length; i++) {
69         // extraemos las 2 primeras cifras del elemento, por lo tanto nos muestras las 2 cifras de la hora del elemento html
70         const acortado = labelBoton[i].innerHTML.substring(0, 2)
71         // si acortado es menor o igual a hora actual (reserva no disponible)
72         if (acortado <= horaActual) {
73           labelBoton[i].style.backgroundColor = "#FD6161";
74           labelBoton[i].style.color = "white";
75         }
76       }
77       //CAMBIAR DISABLED A LOS INPUT QUE YA HA PASADO LA HORA
78       grupo.forEach((element) => {
79         // extraemos las 2 primeras letras/cifras del elemento
80         const values = element.defaultValue.substring(0, 2)
81
82         // si values es menos o igual a la horaActual(reserva no disponible), por lo tanto se pone el elemento disabled
83         if (values <= horaActual) {
84           element.disabled = true;
85         }
86       })
87     }
88     // si la fecha actual es diferente a la fecha del elemento del html
89   } else if ((diaActual != fecha)) {
90     for (var i = 0; i < labelBoton.length; i++) {
91       // extraemos las 2 primeras letras/cifras del elemento
92       const acortado = labelBoton[i].innerHTML.substring(0, 2)
93       // si el elemento de html, es menor o igual a la hora actual (reserva disponible)
94       if (acortado <= horaActual) {
95         labelBoton[i].style.backgroundColor = "";
96         labelBoton[i].style.color = "";
97       }
98     }
99     // QUITAR DISABLE A LOS INPUTS
100    grupo.forEach((element) => {
101      const values = element.defaultValue.substring(0, 2)
102
103      if (values <= horaActual) {
104        //console.log(values)
105        element.disabled = false;
106      }
107    })
108  }
109 }

```

Una vez hayamos realizado la reserva, nos llegará al email un mensaje de confirmación de reserva.



```
@PostMapping("/reservar")
public String reservarPost(@ModelAttribute ReservaDTO reservaDTO, Model model, RedirectAttributes attribute) {

    // Creamos un objeto reserva y le introducimos los datos
    Reserva reserva = new Reserva();
    reserva.setUsuario(reservaDTO.getId_usuario());
    reserva.setPista(reservaDTO.getId_pista());
    reserva.setFecha(reservaDTO.getFecha());
    reserva.setHora_inicio(reservaDTO.getHora_inicio());

    // cambiamos el formato a la fecha
    DateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
    String fechaFormat = formatter.format(reservaDTO.getFecha());

    SimpleMailMessage email = new SimpleMailMessage();

    // si el insert de la reserva es nula nos muestra un error
    if (reservaService.insertReserva(reserva) == null) {
        return "redirect:/reservar?error=Existe&Reserva=";
    }

    // si el insert de reserva no es nulo:
    // se crea la reserva (inserta)
    // enviamos un correo al email de usuario
    email.setTo(reservaDTO.getId_usuario().getEmail());
    email.setSubject("Reserva confirmada.");
    email.setText("Estimado/a " + reservaDTO.getId_usuario().getNombre() + " "
        + reservaDTO.getId_usuario().getApellido1() + " " + reservaDTO.getId_usuario().getApellido2()
        + ", \nle confirmamos su reserva de la pista '" + reservaDTO.getId_pista().getNombre() + "' ("
        + reservaDTO.getId_pista().getDeporte() + ")'" + " para el día " + fechaFormat
        + " con el siguiente tramo horario: " + reservaDTO.getHora_inicio()
        + ". \nGracias por usar nuestros servicios. \nReservaLaPista");

    mailSender.send(email);

    // no muestra mensaje de confirmación en pantalla de 'Mis reservas'
    attribute.addFlashAttribute("success", "Reserva realizada con éxito.");
    // nos redirecciona a la pantalla de 'Mis reservas'
    return "redirect:/misreservas";
}
```

Mis Reservas. Cuando realizamos la reserva, si todo ha ido bien, nos redireccionará a la pantalla de mis reservas, en la cual podemos ver todas las reserva que hemos realizado. Solo podremos cancelar las reservas que aún no se han jugado.

Reserva realizada con éxito.

PDF Excel Copiar

Buscar:

Id	Pista	Fecha	Tramo horario	Estado	Eliminar
84	Padel - Pista Central	11/12/2022	12:00 - 13:00	JUGADO	
87	Padel - Pista Central	11/12/2022	13:00 - 14:00	JUGADO	
92	Padel - Pista Central	11/12/2022	21:00 - 22:00	POR JUGAR	

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Gracias a la implementación de [Datatable](#), podemos descargar nuestros datos de reservas en un archivo PDF, Excel o copiarlo al portapapeles.

Reservas - ReservaLaPista

Id	Pista	Fecha	Tramo horario	Estado
84	Padel - Pista Central	11/12/2022	12:00 - 13:00	JUGADO
87	Padel - Pista Central	11/12/2022	13:00 - 14:00	JUGADO
93	Padel - Pista Central	11/12/2022	21:00 - 22:00	POR JUGAR

Mostrar todo

Autoguardado Reservas - ReservaLaPista.xlsx - Vista protegida ▾

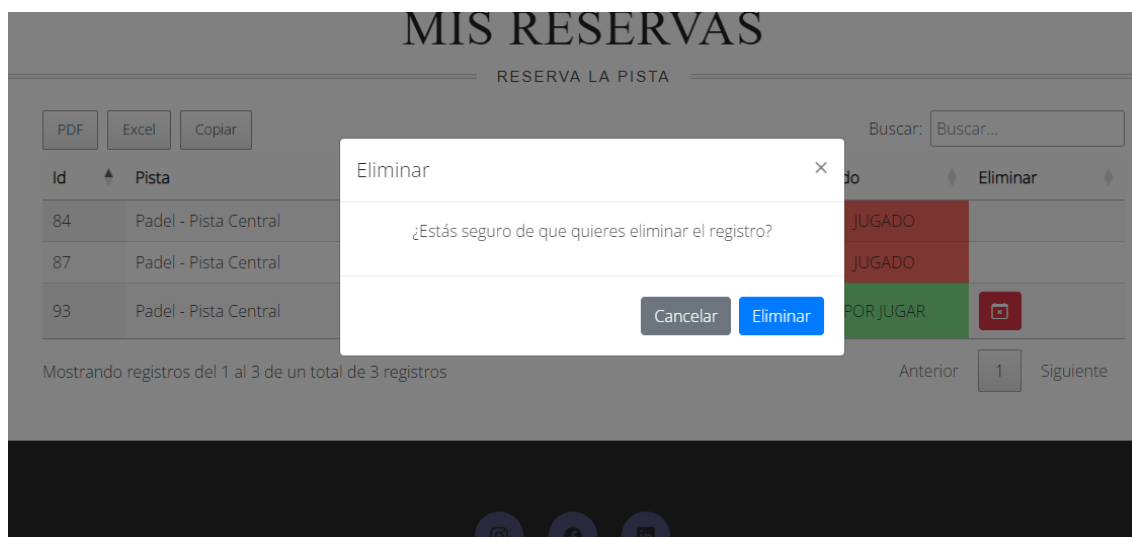
Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vis

VISTA PROTEGIDA Tenga cuidado: los archivos de Internet pueden contener virus. Si no tiene que editarlo,

A1 Reservas - ReservaLaPista

	A	B	C	D	E	F
1	Reservas - ReservaLaPista					
2	Id	Pista	Fecha	Tramo horario	Estado	
3	84	Padel - Pista Central	11/12/2022	12:00 - 13:00	JUGADO	
4	87	Padel - Pista Central	11/12/2022	13:00 - 14:00	JUGADO	
5	93	Padel - Pista Central	11/12/2022	21:00 - 22:00	POR JUGAR	
6						
7						

Para eliminar la reserva, nos pedirá que confirmemos.



Cuando confirmamos, recibiremos un correo con la cancelacion de la reserva.

Reserva cancelada. Externo Recibidos x



reservalapista2022@gmail.com

para mí ▾

Estimado/a Usuario Apellido1 Apellido2,
 le confirmamos que su reserva ha sido eliminada.
 Pista Central --- Padel
 11/12/2022 --- 21:00 - 22:00
 Gracias por usar nuestros servicios.
 ReservaLaPista

Y se nos mostrará un mensaje de cancelacion en la pantalla de “Mis reservas”.

MIS RESERVAS

RESERVA LA PISTA

Su reserva se ha cancelado. ×

Buscar:

Id	Pista	Fecha	Tramo horario	Estado	Eliminar
84	Padel - Pista Central	11/12/2022	12:00 - 13:00	JUGADO	
87	Padel - Pista Central	11/12/2022	13:00 - 14:00	JUGADO	

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior
1
Siguiente

Ajustes. En la pantalla de ajustes podremos editar nuestros datos , cambiar la contraseña y eliminar la cuenta.

Ajustes
 x +

reservalapista.up.railway.app/ajustes
 🔍 🌐 ⭐

ReservaLaPista 20:25
Inicio Reserva de Espacio Usuario Apellido1 Apellido2

Mis Datos
Cambiar Contraseña
Eliminar Cuenta

Id: 15

Email: alum.sbermudezh@iesalixar.org

DNI: 00000002A

Usuario: usuario

Role: ROLE_USER

Fecha de Registro: 04/12/2022

Nombre:

Segundo apellido:

Provincia:

Sexo:

Primer apellido:

Localidad:


Telefono:

Fecha de nacimiento:

GUARDAR

La contraseña estará segura porque está encriptada, como en todos los registros de contraseña de la aplicación, como por ejemplo en el alta de usuario.

Mis Datos **Cambiar Contraseña** Eliminar Cuenta

Nueva contraseña 

GUARDAR

```
// creamos el usuario y le introducimos los datos
Usuario usuario = new Usuario();
usuario.setId(usu.getId());

usuario.setPassword(new BCryptPasswordEncoder(15).encode(usu.getPassword()));
usuario.setReserva(re);

// si el tamaño de la contraseña es menor de 5 nos mostrara un error
```

Por último, podremos eliminar nuestra cuenta. También recibiremos un correo confirmando que nos hemos dado de baja. Después de eliminar la cuenta se nos cerrará la sesión y nos devuelve al login principal.

Mis Datos Cambiar Contraseña **Eliminar Cuenta**

BORRAR MI CUENTA

Confirmación cuenta eliminada. **Externo**



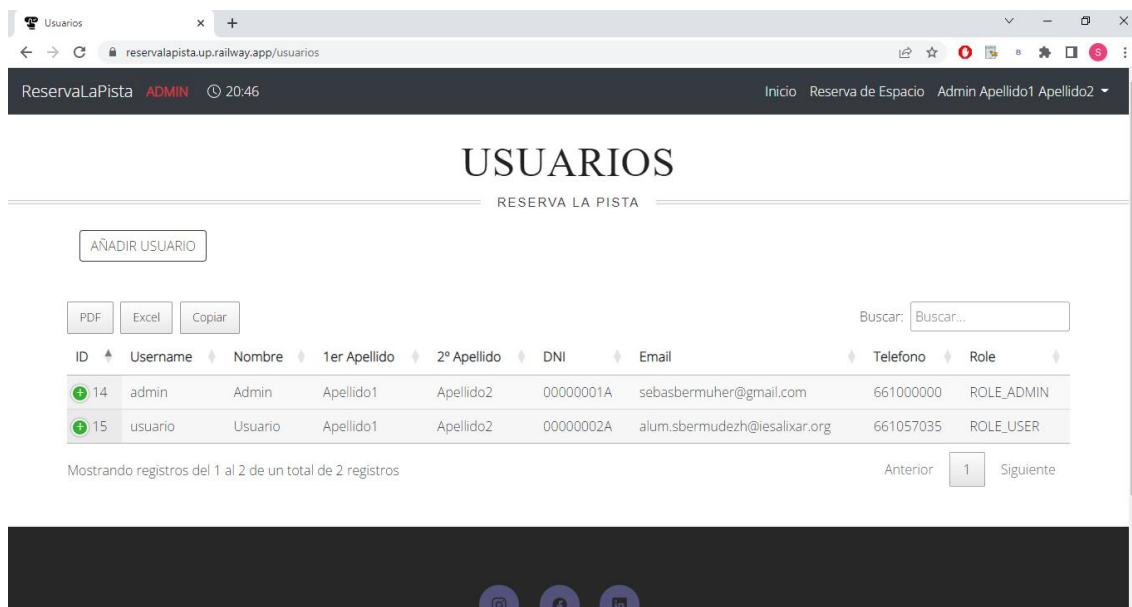
reservalapista2022@gmail.com
para mí ▼

Estimado/a Usuario Apellido1 Apellido2,
le confirmamos que su cuenta de ReservaLaPista ha sido eliminada.
Gracias por usar nuestros servicios.
ReservaLaPista

7.- Manual de administrador.

Aquí se incluyen las pantallas “extras” solo para el usuario con rol = role_admin.

Usuarios. En esta pantalla se muestra todos los usuarios registrados en la aplicación. Tiene varias opciones de gestión.



El administrador puede crear una nueva cuenta. Una vez la cree, nos redirige a la pantalla de los usuarios y nos muestra un mensaje de confirmación.

El usuario registrado recibirá un email de confirmación.

Debemos tener en cuenta que los usuarios tienen 3 claves únicas (username, DNI y email), por lo tanto, ninguna de estas 3 pueden estar repetidas en la base de datos. Lo cual, de intentar duplicar algún registro, nos devolverá un error.

AÑADIR USUARIO

RESERVA LA PISTA

Nombre

Primer apellido

Segundo apellido

Usuario

Contraseña

Email

Role

ROLE_USER

DNI

Localidad

Provincia

Telefono

Sexo

Hombre

Fecha de nacimiento

dd/mm/aaaa

AÑADIR

Tenemos varias opciones sobre la gestión de cada usuario.

AÑADIR USUARIO

PDF Excel Copiar

Buscar: Buscar...

ID	Username	Nombre	1er Apellido	2º Apellido	DNI	Email	Telefono	Role	Fecha Nacimiento	Sexo
14	admin	Admin	Apellido1	Apellido2	00000001A	sebasbermuher@gmail.com	661000000	ROLE_ADMIN	10/07/1999	Hombre
15	usuario	Usuario	Apellido1	Apellido2	00000002A	alum.sbermudezh@iesalixar.org	661057035	ROLE_USER	10/07/1999	Hombre

Fecha Registro 04/12/2022

Localidad Tomares

Provincia Sevilla

Info

Reservas

Editar

Eliminar

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Ver la información del usuario, donde nos saldrá todos sus datos.

Info de usuario

reservalapista.up.railway.app/usuarios/info?user=15

ReservaLaPista ADMIN 20:50 Inicio Reserva de Espacio Admin Apellido1 Apellido2

INFORMACIÓN

@USUARIO

[VOLVER ATRÁS](#)

Id: 15	Username: usuario	Nombre: Usuario
1er Apellido: Apellido1	2º Apellido: Apellido2	Email: alum.sbermudezh@iesalixar.org
DNI: 00000002A	Telefono: 661057035	Sexo: Hombre
Role: ROLE_USER	Fecha Nacimiento: 10/07/1999	Fecha Registro: 04/12/2022
Localidad: Tomares	Provincia: Sevilla	

Inicio Sobre Nosotros Horarios Ubicación Cerrar Sesión

Ver las reservas del usuario seleccionado, con la opción de eliminar cualquiera de sus reservas, jugada y sin jugar. Si la eliminamos, se le notificara al usuario por email.

Reservas de usuario

reservalapista.up.railway.app/usuarios/reservas?codigo=15

ReservaLaPista ADMIN 20:50 Inicio Reserva de Espacio Admin Apellido1 Apellido2

RESERVAS

@USUARIO

PDF Excel Copiar

Buscar:

Id	Fecha	Tramo horario	Pista	Estado	
84	11/12/2022	12:00 - 13:00	Padel - Pista Central	JUGADO	
87	11/12/2022	13:00 - 14:00	Padel - Pista Central	JUGADO	

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Inicio Sobre Nosotros Horarios Ubicación Cerrar Sesión

Editar el usuario seleccionado.

Editar Usuario

reservalapista.up.railway.app/usuarios/edit?user=15

ReservaLaPista ADMIN 20:52 Inicio Reserva de Espacio Admin Apellido1 Apellido2

EDITAR USUARIO

@USUARIO

Id: 15

Nombre: Usuario

Primer apellido: Apellido1

Segundo apellido: Apellido2

Usuario: usuario

Email: alum.sbermudezh@iesalixar.org

Role: ROLE_USER

DNI: 00000002A

Localidad: Tomares

Provincia: Sevilla

Telefono: 661057035

Sexo: Hombre

Fecha de nacimiento: 10/07/1999

Fecha de registro: 04/12/2022

EDITAR

Y eliminar usuario. También se le notificará al usuario que se ha eliminado su cuenta.

Usuarios

reservalapista.up.railway.app/usuarios#

AÑADIR USUARIO

PDF Excel Copiar

Buscar: Buscar...

ID	Username	Nombre	1er Apellido	2º Apellido	DNI	Email	Telefono	Role	Fecha Nacimiento	Sexo
14	admin	Admin	Apellido1	Apellido2	00000001A	sebasbermuher@gmail.com	661000000	ROLE_ADMIN	10/07/1999	Hombre
15	usuario	Usuario	Apellido1	Apellido2	00000002A	alum.sbermudezh@iesalixar.org	661057035	ROLE_USER	10/07/1999	Hombre

Fecha Registro: 04/12/2022

Localidad: Tomares

Provincia: Sevilla

Info

Reservas

Editar

Eliminar

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Eliminar

¿Estás seguro de que quieres eliminar el registro?

Cancelar Eliminar

PISTAS. En esta pantalla visualizaremos todas las pantallas que tenemos registradas en la aplicación, así como sus datos. Todo esto en una tabla.

También tenemos varias opciones de gestión, como por ejemplo, añadir, editar, ver reservas de una pista o eliminar pista.

Se ha añadido una nueva pista. "Pista Lateral" guardada con éxito.

AÑADIR PISTA

PDF Excel Copiar

Buscar:

ID	Nombre	Deporte	Apertura	Cierre	Editar	Reservas	Eliminar
42	Pista Central	Padel	10:00	22:00			
45	Futbol 7	Futbol	10:00	22:00			
46	Pista Lateral	Padel	10:00	22:00			

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Para añadir pista solo debemos de introducir la siguiente información. Debemos tener en cuenta que no podemos llamar a dos pistas con el mismo nombre, ya que en la base de datos la tenemos como una clave única. En este caso nos devolvería un error.

AÑADIR PISTA

RESERVA LA PISTA

Nombre

Deporte

Hora de apertura

Hora de cierre

AÑADIR

Inicio Sobre Nosotros Horarios Ubicación Cerrar Sesión

Para editar pista podremos modificar cualquiera de sus datos, aunque debemos tener cuidado de no llamar a dos pistas iguales.

Editar Pista

reservapista.up.railway.app/pistas/edit?pist=42

ReservaLaPista ADMIN 21:01 Inicio Reserva de Espacio Admin Apellido1 Apellido2

EDITAR

PISTA CENTRAL

Id: 42

Nombre: Pista Central

Deporte: Padel

Hora de apertura: 10:00

Hora de cierre: 22:00

EDITAR

Aquí podemos ver todas las reservas que tiene una pista. Podremos eliminar cada una de sus reservas.

Tenemos la opción de generar un PDF o Excel, o si lo preferimos copiarlo al portapapeles.

Reservas de pista

reservapista.up.railway.app/pistas/reservas?codigo=42

ReservaLaPista ADMIN 21:02 Inicio Reserva de Espacio Admin Apellido1 Apellido2

RESERVAS

Pista Central (PADEL)

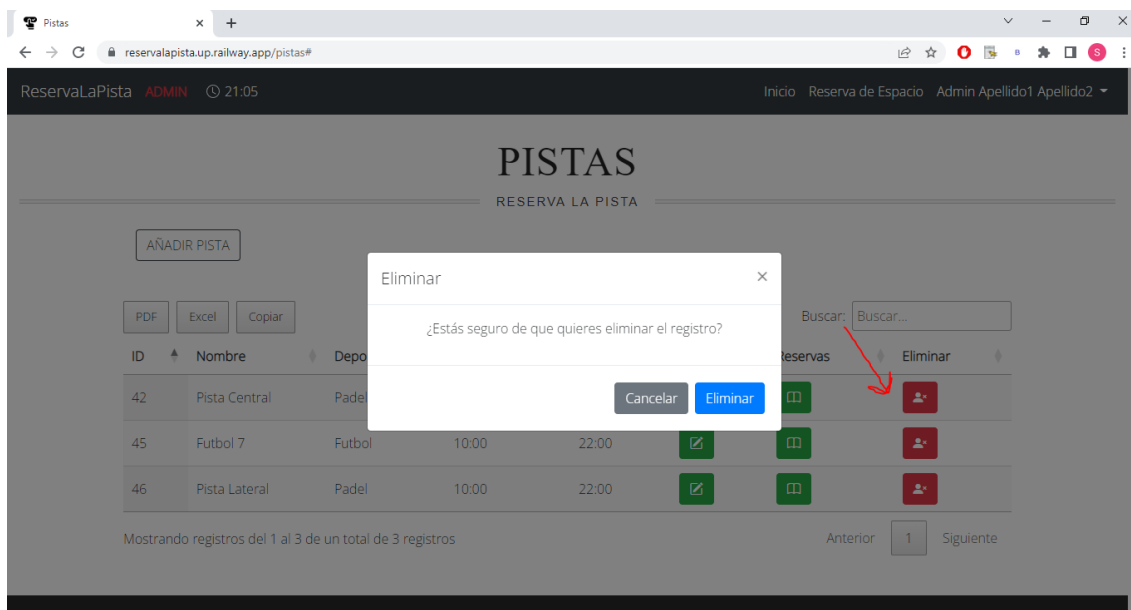
VOLVER ATRÁS

PDF Excel Copiar

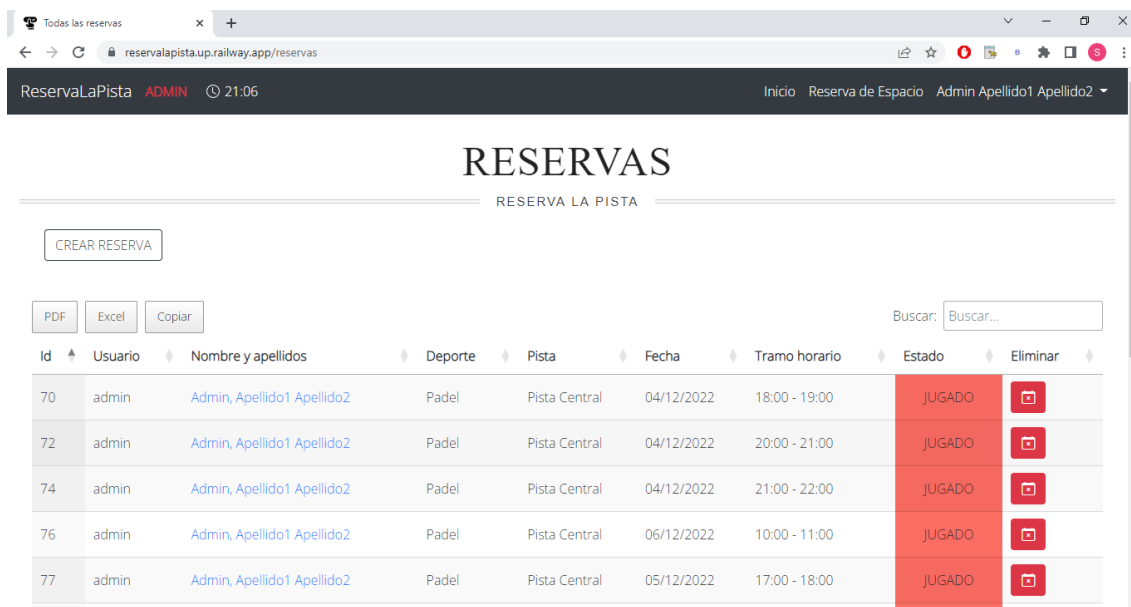
Buscar: Buscar...

Id	Usuario	Fecha	Tramo horario	Estado	Eliminar
70	admin	04/12/2022	18:00 - 19:00	JUGADO	
72	admin	04/12/2022	20:00 - 21:00	JUGADO	
74	admin	04/12/2022	21:00 - 22:00	JUGADO	
76	admin	06/12/2022	10:00 - 11:00	JUGADO	
77	admin	05/12/2022	17:00 - 18:00	JUGADO	
78	admin	10/12/2022	11:00 - 12:00	JUGADO	

Por último, podremos eliminar la pista confirmando el mensaje



RESERVAS. Aquí podremos ver todas las reservar que tiene la web con sus diferentes estados y sus opciones de gestión.



Para los estados básicamente se coge la hora actual y el tramo de horario y se establece un if para saber en la situación que se encuentra.

```

1 // creamos un objeto día con la fecha actual
2 let dia = new Date()
3 // extraemos solo el día de la fecha (día)
4 let diaActual = dia.toISOString().substring(0, 10)
5 // extraemos la hora y los minutos de la fecha actual (ej-> 12:15)
6 let horaActual = dia.toLocaleString().split(',')[1].substring(1, 6)
7
8 // recogemos los siguientes elementos del documento html
9 let fechaDOM = document.getElementsByName("fecha")
10 let horas = document.getElementsByName("horas")
11 let estado = document.getElementsByName("estado")
12
13 // de 0 hasta el tamaño de fechaDOM (recorre todo 'fechaDOM')
14 for (var i = 0; i < fechaDOM.length; i++) {
15     // nos devuelve todos los valores del elemento, en este caso las fechas (con name="fecha")
16     const fecha = fechaDOM[i].innerHTML
17     // nos devuelve la parte de las horas de las horas de inicio
18     const horasInicio = horas[i].innerHTML.substring(0, 5)
19     // nos devuelve la parte de las horas de las horas fin
20     const horasFin = horas[i].innerHTML.substring(8, 13)
21     // separamos la fecha
22     const año = fecha.split('-')[0]
23     const mes = fecha.split('-')[1]
24     const dia = fecha.split('-')[2]
25
26     // si el elemento fecha recogido del elemento es menor al día actual (por lo tanto el partido ya se ha jugado)
27     if (fecha < diaActual) {
28         estado[i].innerHTML = "JUGADO"
29         estado[i].style.backgroundColor = "#FB6C63"
30         estado[i].style.textAlign = "center"
31     }
32     // si el elemento fecha recogido del elemento es mayor al día actual (aun no se ha jugado el partido)
33     if (fecha > diaActual) {
34         estado[i].innerHTML = "POR JUGAR"
35         estado[i].style.backgroundColor = "#85EA8E"
36         estado[i].style.textAlign = "center"
37     }
38 }
39 // si el elemento fecha recogido del elemento es igual al día actual
40 if (fecha === diaActual) {
41     // si la hora actual es mayor o igual, el partido esta en juego
42     if (horaActual >= horasInicio) {
43         estado[i].innerHTML = "EN JUEGO"
44         estado[i].style.backgroundColor = "#FBF663"
45         estado[i].style.textAlign = "center"
46         // si la hora actual es mayor a la hora fin, el partido ya se ha jugado
47         if (horaActual > horasFin) {
48             estado[i].innerHTML = "JUGADO"
49             estado[i].style.backgroundColor = "#FB6C63"
50             estado[i].style.textAlign = "center"
51         }
52     }
53     } else {
54         // si la hora actual es menor a la ahora inicio, el partido aun no se ha jugado
55         estado[i].innerHTML = "POR JUGAR"
56         estado[i].style.backgroundColor = "#85EA8E"
57         estado[i].style.textAlign = "center"
58     }
59 }
60 // formateo de fecha (dd/mm/yyyy)
61 fechaDOM[i].innerHTML = dia + "/" + mes + "/" + año
62 }

```

El administrador puede crear una reserva para cualquier usuario. Para el tramo horario tiene las mismas restricciones que para cualquier usuario.

Una vez creada la reserva, el usuario recibirá un email de confirmación.

Si en la tabla de reservas hacemos click en el nombre de usuario, podremos ver todos sus datos.

RESERVAS									
RESERVA LA PISTA									
CREAR RESERVA									
PDF		Excel		Copiar		Buscar: <input type="text" value="Buscar..."/>			
Id	Usuario	Nombre y apellidos	Deporte	Pista	Fecha	Tramo horario	Estado	Eliminar	
70	admin	Admin, Apellido1 Apellido2	Padel	Pista Central	04/12/2022	18:00 - 19:00	JUGADO		
72	admin	Admin, Apellido1 Apellido2	Padel	Pista Central	04/12/2022	20:00 - 21:00	JUGADO		

The screenshot shows the 'Información' page for user 'ADMIN'. The page displays a grid of user details:

Id: 14	Username: admin	Nombre: Admin
1er Apellido: Apellido1	2º Apellido: Apellido2	Email: sebasbermuher@gmail.com
DNI: 00000001A	Telefono: 661000000	Sexo: Hombre
Role: ROLE_ADMIN	Fecha Nacimiento: 10/07/1999	Fecha Registro: 04/12/2022
Localidad: Tomares	Provincia: Sevilla	

At the bottom of the page, there are social media icons for Instagram, Facebook, and LinkedIn.

Tenemos la opción de eliminar cualquier reserva, haya pasado ya su fecha y hora o no.

Para eliminar una reserva debemos confirmar el mensaje, y el usuario de la reserva recibirá una notificación en el email.

The screenshot shows the 'Reservas' page with a confirmation dialog box titled 'Eliminar' (Delete). The dialog asks: '¿Estás seguro de que quieres eliminar el registro?' (Are you sure you want to delete the record?). There are 'Cancelar' (Cancel) and 'Eliminar' (Delete) buttons.

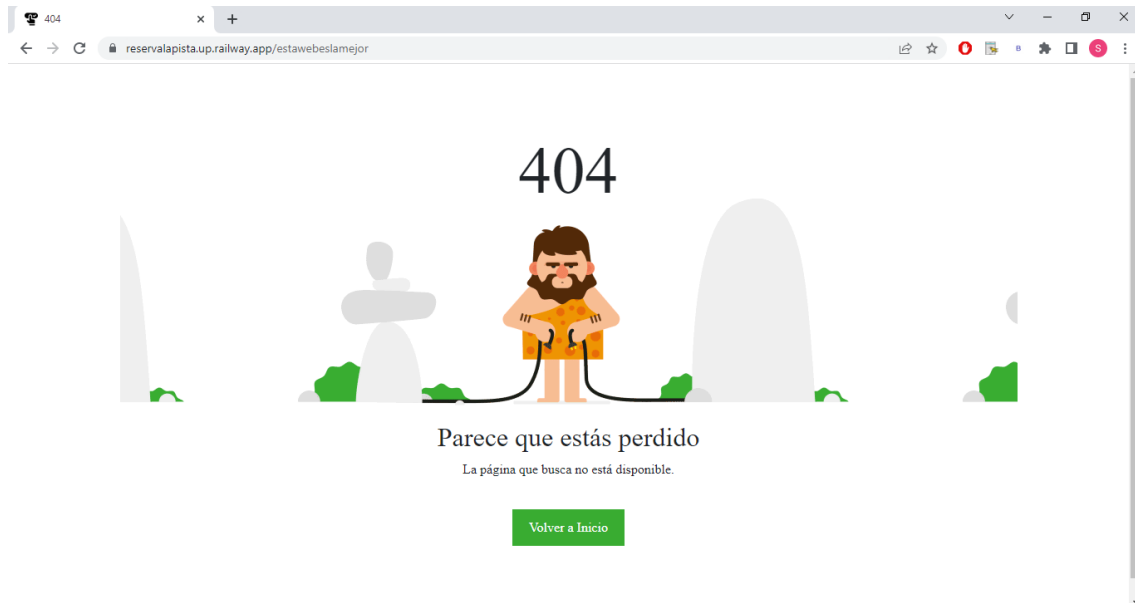
The background shows a table of reservations:

Id	Usuario	Nombre y apellidos	Estado	Eliminar
70	admin	Admin, Apellido1 Apellido2	JUGADO	[Icon]
72	admin	Admin, Apellido1 Apellido2	JUGADO	[Icon]
74	admin	Admin, Apellido1 Apellido2	JUGADO	[Icon]
76	admin	Admin, Apellido1 Apellido2	JUGADO	[Icon]
77	admin	Admin, Apellido1 Apellido2	JUGADO	[Icon]

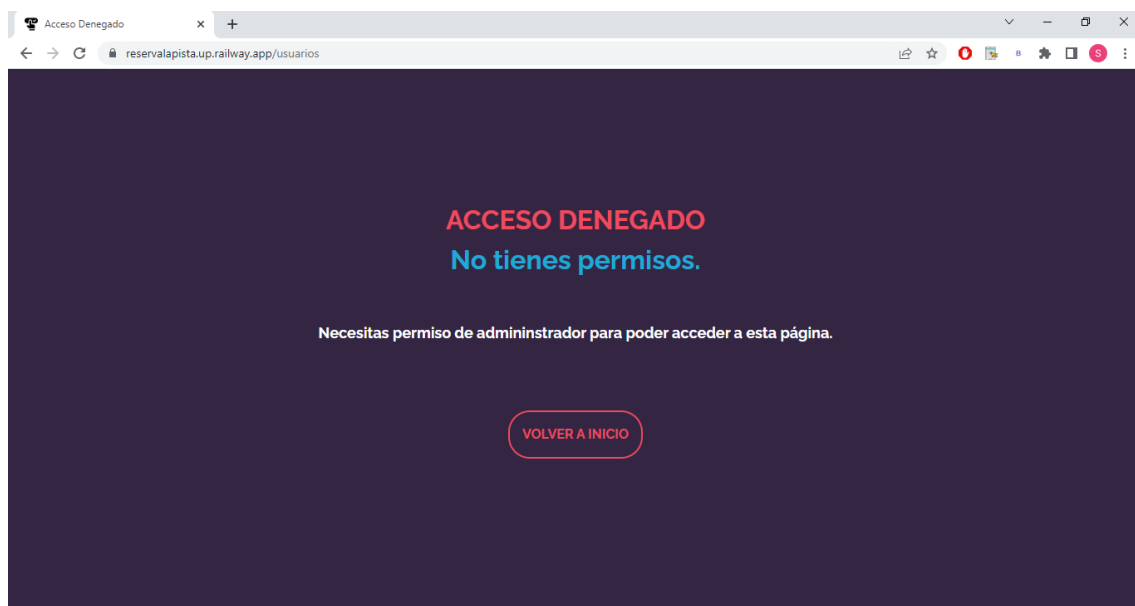
ERRORES PERSONALIZADOS.

Se ha personalizado las siguientes pantallas de errores.

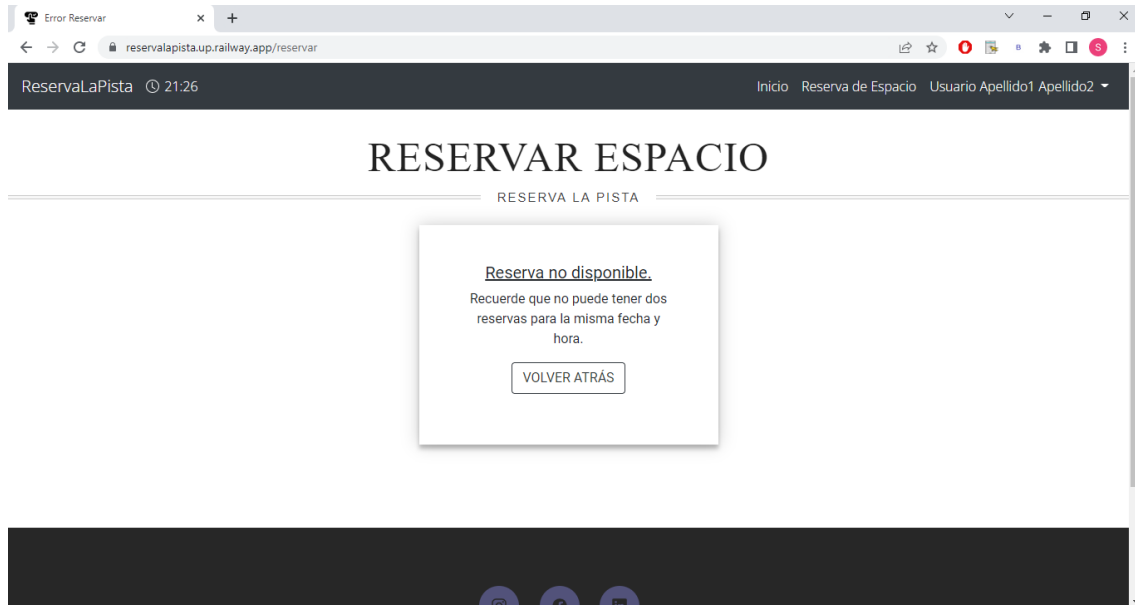
Not found. Error comúnmente llamado 404. Este error nos suele aparecer cuando intentamos navegar a una pagina de nuestro sitio web que no existe.



Forbidden. Cuando un usuario no tiene permisos. Por ejemplo, el usuario con rol_user no tiene permiso para acceder a la pantalla de /usuarios.



Error Server 500. Error por parte del servidor. Por ejemplo, cuando una reserva no está disponible por que ya está reservada por otro usuario, o porque un usuario quiere realizar una reserva y ya tiene otra confirmada para la misma fecha y hora.



Archivo *application.properties*

```
server.error.whitelabel.enabled=false
server.error.path=/error
```

```
1 package org.iesalixar.servidor.controller;
2
3 import java.security.Principal;
4
5 @Controller
6 public class MyErrorController implements ErrorController {
7     @Autowired
8     UsuarioServiceImpl usuarioService;
9
10    // -----
11    // PANTALLAS ERRORES
12    // -----
13    // Pantallas de errores personalizada
14    @RequestMapping("/error")
15    public String handleError(HttpServletRequest request, Model model, Principal principal) {
16        // Para mostrar nombre y apellidos del usuario que ha iniciado sesion
17        Usuario user = usuarioService.getUsuarioByUserName(principal.getName());
18        model.addAttribute("user", user);
19        // -----
20        Object status = request.getAttribute(RequestDispatcher.ERROR_STATUS_CODE);
21
22        if (status != null) {
23            Integer statusCode = Integer.valueOf(status.toString());
24
25            if (statusCode == HttpStatus.NOT_FOUND.value()) {
26                return "error-404";
27            } else if (statusCode == HttpStatus.INTERNAL_SERVER_ERROR.value()) {
28                return "error-500";
29            } else if (statusCode == HttpStatus.FORBIDDEN.value()) {
30                return "forbidden";
31            }
32        }
33        return "error-404";
34    }
35 }
```

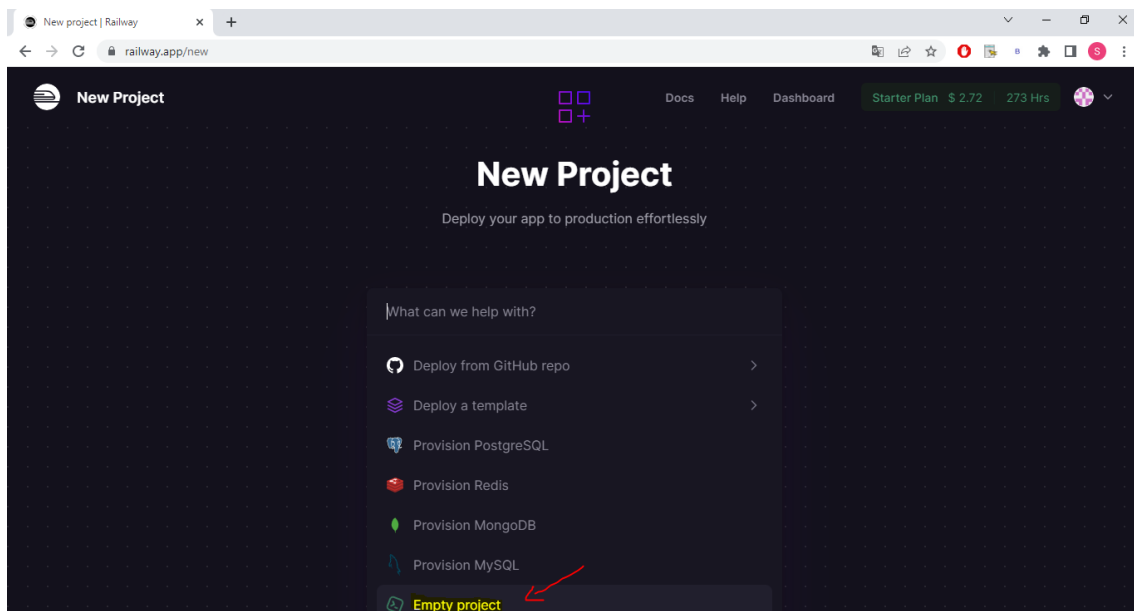
8.- Manual de despliegue.

El proyecto lo he desplegado en la web [Railway](#), en la cual podemos conectar nuestra cuenta Github y enlazar nuestros repositorios.

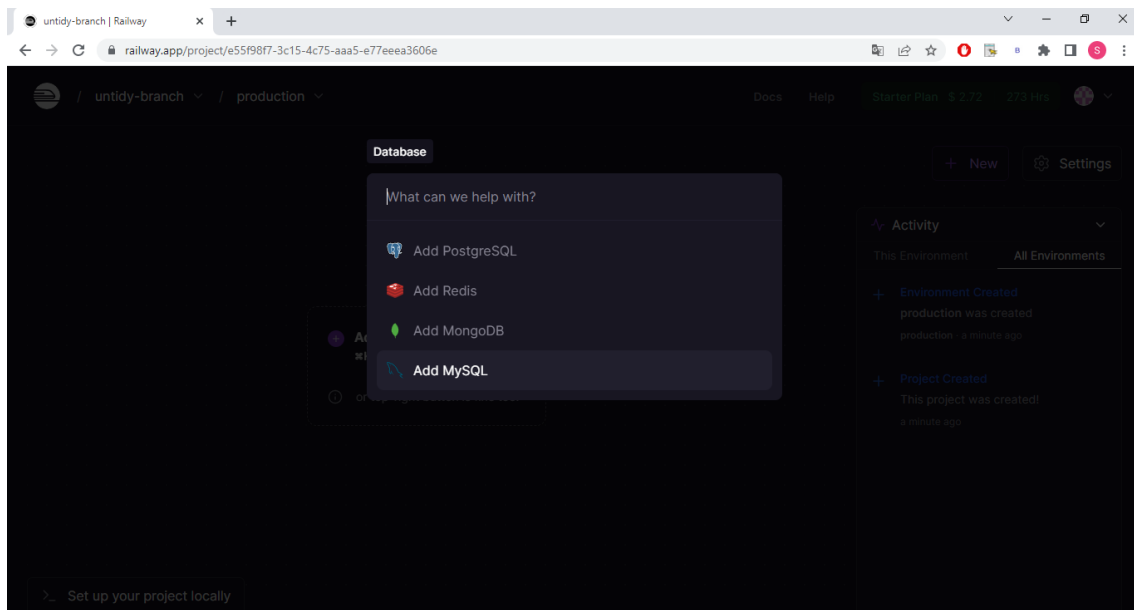
Las subidas son muy fáciles ya que función actualizando nuestro repositorio con un ‘push’.

Pero antes de todo esto, hay que realizar la configuración, desde la propia web de Railway.

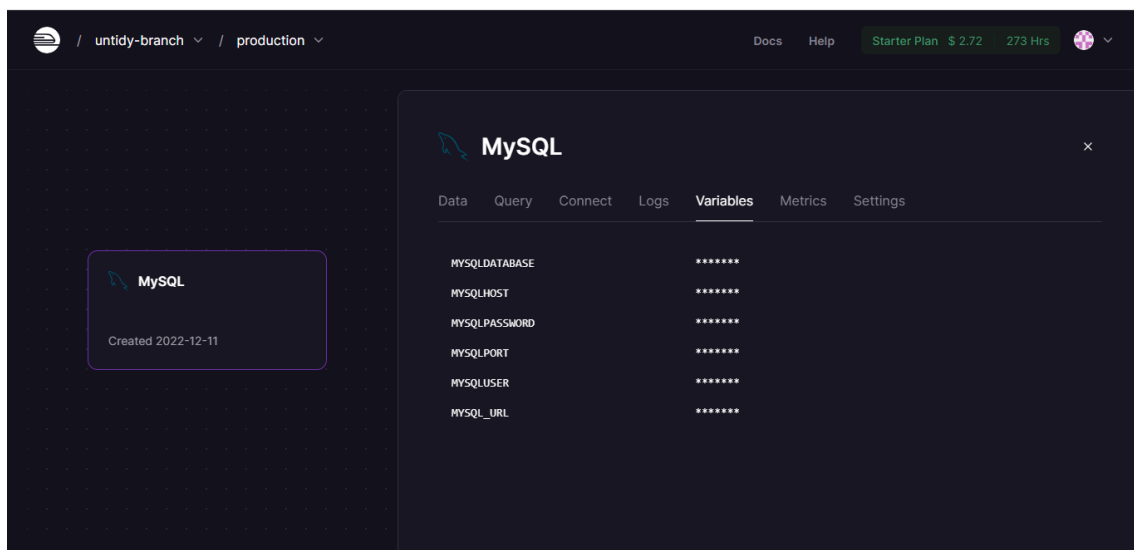
Creamos un nuevo proyecto. En este caso será un “Empty project”.



Le añadimos un nuevo servicio MySQL.



Vemos las variables que nos crea.



Y la introducimos en el *applications.properties* de nuestra aplicación.

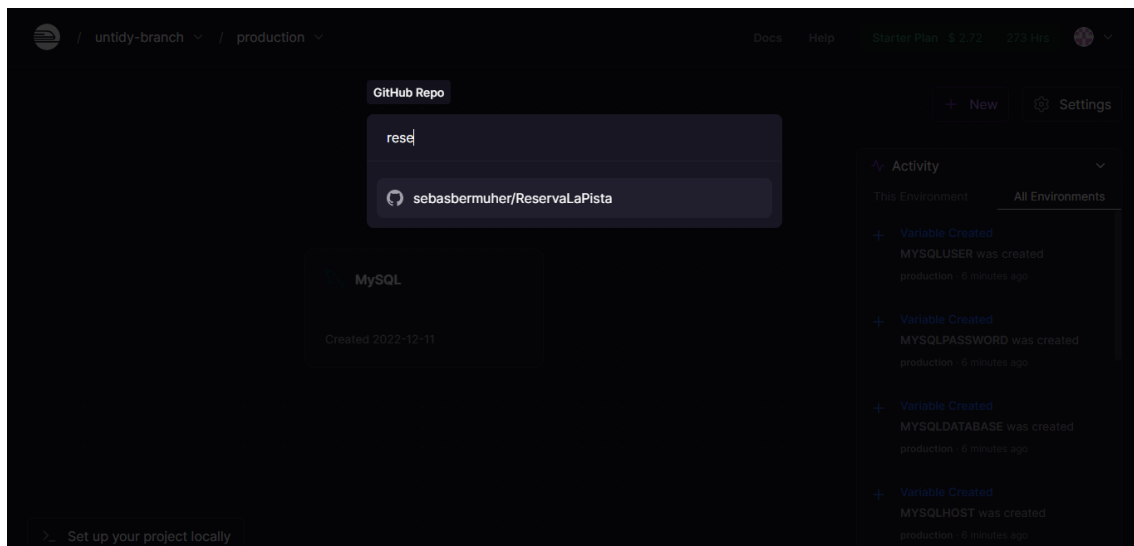
```

15
16 #SERVIDOR
17 spring.datasource.url=jdbc:mysql://root:xR2uAL9tsrnyKdSFEYp@containers-us-west-140.railway.app:7248/railway?createDatabaseIfNotExist=tr
18 spring.datasource.username=root
19 spring.datasource.password=xR2uAL9tsrnyKdSFEYp
20 spring.jpa.hibernate.ddl-auto=update
21 spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
22 spring.jpa.show-sql=true
23 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
24 #spring.jpa.generate-ddl=update
  
```

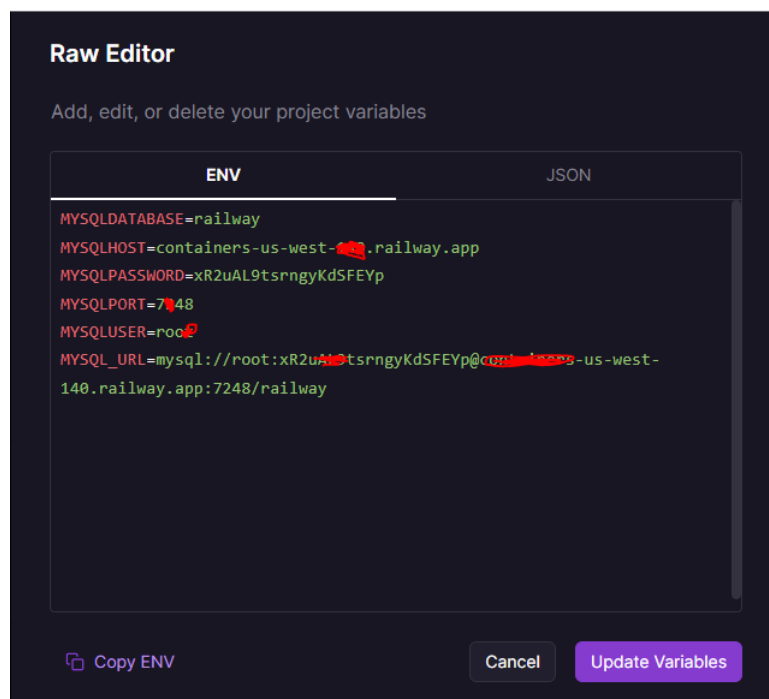

En mi caso, al estar trabajando en local con MariaDB, debo de cambiar de dependencias en el archivo pom.xml.

```
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
</dependency>
<dependency>
```

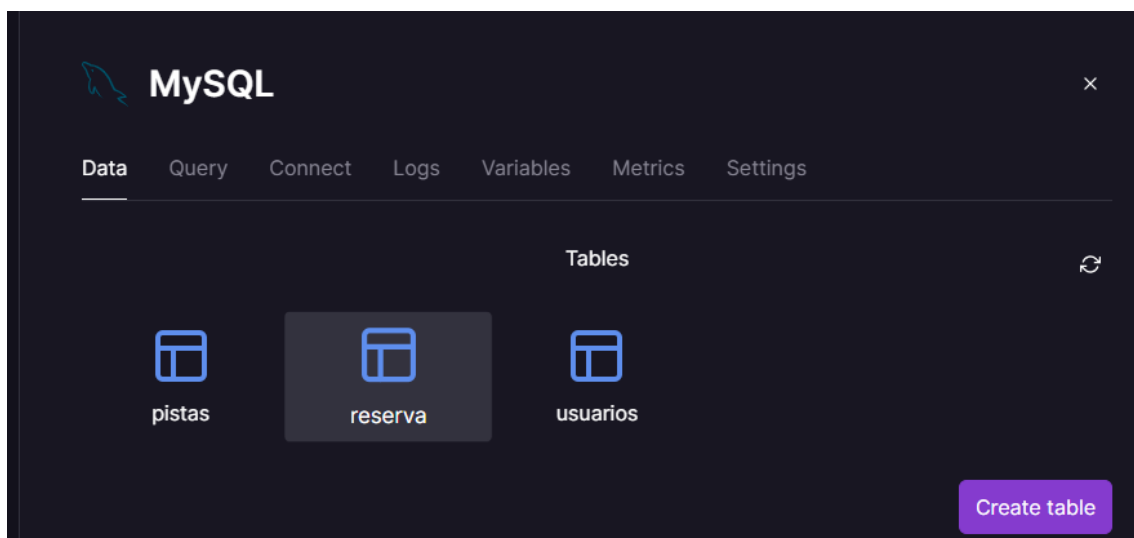
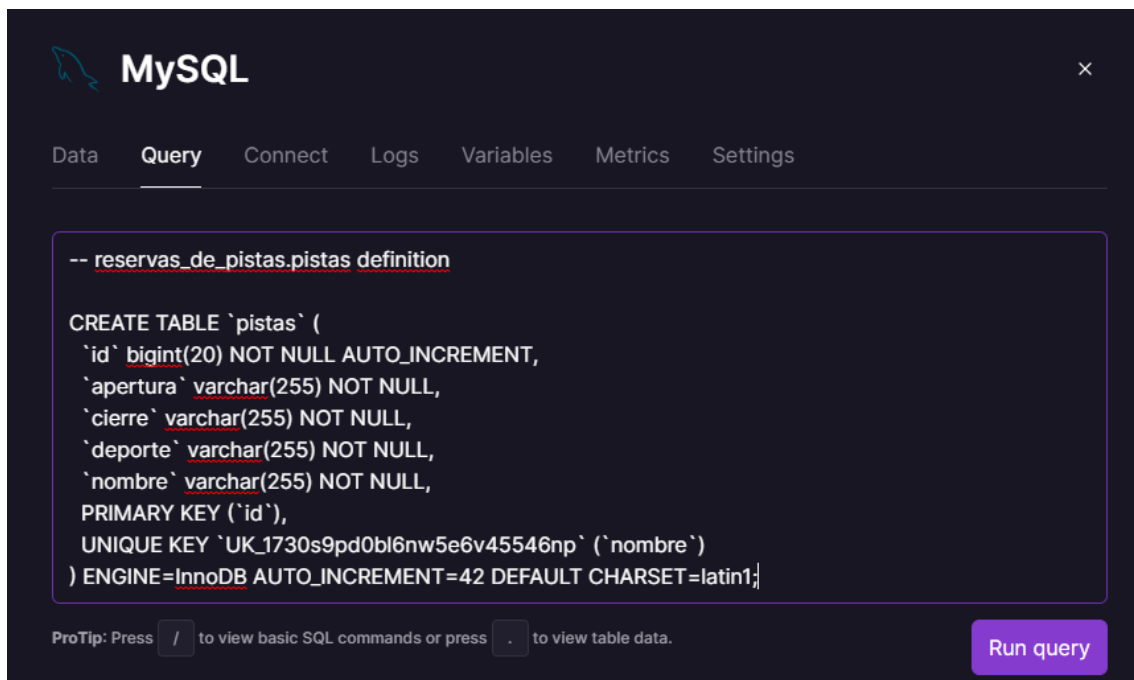
Añado un nuevo servicio. En este caso debe ser el repositorio de GitHub.



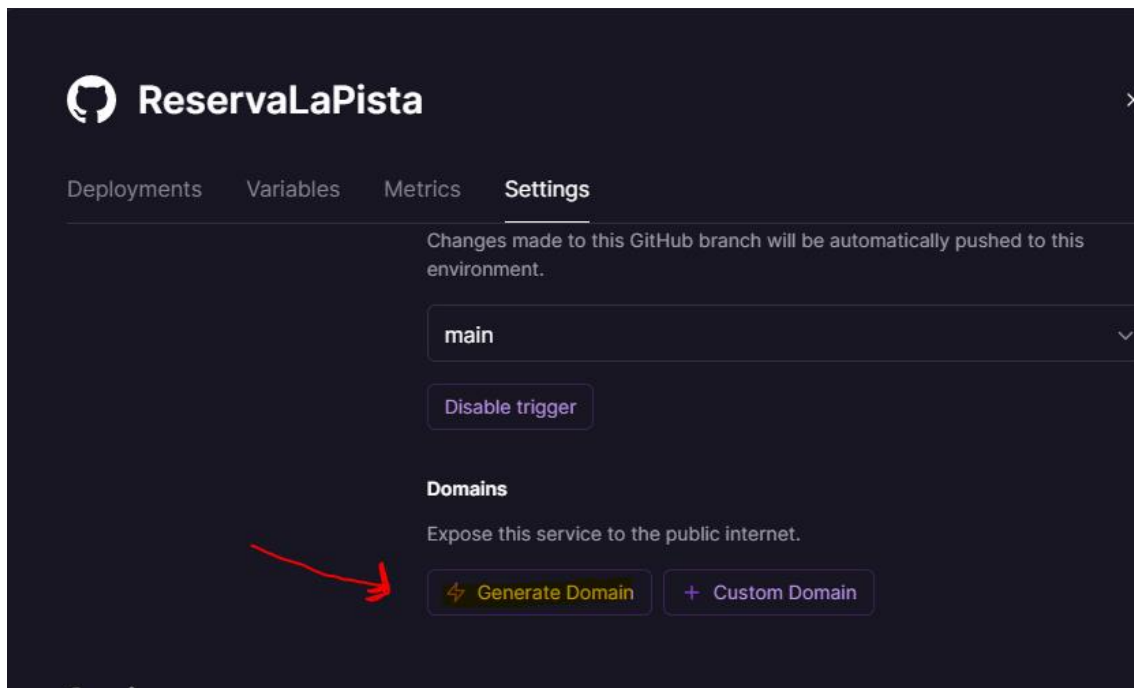
Y le introducimos las variables SQL anteriores.



En el servicio MySQL creamos las tablas mediante el código, ya que, al tener muchas claves, si creas las tablas con la interfaz gráfica de este sistema, no tendrás esta opción.



Y generamos un dominio.



9.- Conclusión.

Para concluir pienso que es una buena idea de proyecto a tratar, a la que quizás hay que pulir aún más.

Algunas ideas que podrían mejorar su uso podrían ser:

- La implementación de forma de pago.
- Que la propia aplicación te permita buscar partido con otros rivales.
- Crear un formulario de contacto en el que el usuario rellenando un formulario le envíe un mensaje al email de la empresa.

Pongo fin a este proyecto confirmando que ha sido una gran experiencia para mí, desde la creación de mi primer proyecto personal hasta todos estos cursos del grado, en general.

Ahora veo toda mi evolución desde que me matriculé sin saber casi nada de programación y no saber cómo me iría.

Por ello me siento ORGULLOSO.

Han sido muchas horas incansables picando código y buscando información, pero han valido la pena, puesto que me ha gustado desarrollar este trabajo.

Desde estas humildes líneas me gustaría agradecer a mi familia cuantísimo apoyo que me dan, y a todos los profesores que he tenido por demostrarme sus conocimientos.

Especial agradecimiento a mi tutor de proyecto D. Antonio Gabriel González Casado.