

# Memoria de Trabajo

## Proyecto de Fin de Ciclo

Petland Resort



P E T L A N D   R E S O R T



Autora: Dalieska Andrea Vieira Hernández

Tutor: Prof. Antonio Gabriel González Casado,

Centro: IES Alixar

Ciclo: Desarrollo de Aplicaciones Web

Curso académico: 2022-2023



# Índice

|   |    |
|---|----|
| 1. Introducción   | 2  |
| 2. Objetivos y Motivación                                 | 2  |
| 3. Análisis de las alternativas del mercado               | 3  |
| 4. Tecnologías empleadas                                  | 4  |
| 4.1. Desarrollo   | 5  |
| 4.1.1. Back   | 5  |
| 4.1.2. Front  | 5  |
| 4.2. Despliegue   | 6  |
| 4.3. Útiles   | 6  |
| 4.3.1. Construcción                                       | 6  |
| 4.3.2. Control  | 6  |
| 4.3.3. Diseño   | 6  |
| 4.3.4. Pruebas  | 7  |
| 4.3.5. Otros  | 7  |
| 5. Diseño de la aplicación                                | 7  |
| 5.1. Funcionalidades                                      | 7  |
| 5.2. Diseño de la base de datos                           | 10 |
| 5.2.1. Esquema  | 10 |
| 5.2.2. Modelado de datos                                  | 11 |
| 5.3. Vistas   | 20 |
| 5.3.1. Wireframe  | 20 |
| 5.3.2. Mockup   | 22 |
| 6. Definición API REST: publicación de servicios          | 44 |
| 6.1. Index  | 44 |
| 6.2. Users  | 45 |
| 6.3. Pets   | 46 |
| 6.4. Reservations   | 48 |
| 6.5. Datas  | 50 |
| 7. Manual de despliegue                                   | 51 |
| 7.1. Archivos de configuración                            | 51 |
| 7.2. Configuración de la plataforma de despliegue Railway | 52 |
| 8. Postmortem y conclusiones                              | 59 |



## I. Introducción

En el documento que se presenta a continuación, se muestran las fases que componen el proyecto de fin de curso para el ciclo de Desarrollo de Aplicaciones Web. El proyecto pretende aunar en una pequeña aplicación todos los conocimientos adquiridos a lo largo de este ciclo, haciendo especial énfasis en los módulos del último año.

Para la realización de este proyecto se han tenido en cuenta distintos aspectos como que sea factible en tiempo, forma y contenido; que sea una muestra significativa de los conocimientos; y que responda a una necesidad real.

Se puede consultar el repositorio con toda la información y el proyecto a través de la siguiente url: <https://github.com/dandreavh/PetlandResort-ProyectoFinal>

## 2. Objetivos y Motivación

Son muchos los motivos por los que se ha realizado este proyecto. Por una parte, la motivación tecnológica, por el uso de un stack completo en JavaScript, un lenguaje de programación con gran impacto en la actualidad, una base de datos no relacional y un sistema, en general, de creación de aplicaciones muy rápido, moderno y con mucha flexibilidad.

Por otra parte, la idea de negocio ha sido el motor principal para trabajarla con mimo y detalle, pues mi sueño siempre ha sido tener un espacio para el cuidado de animales; por lo tanto, este proyecto reúne pasión y profesión.

Como objetivo, además de la satisfacción personal de realizar un aplicativo web completo y funcional, se quiere responder a una necesidad de nuestro cliente, un complejo hotelero, Petland Resort, que se encarga de alojar mascotas y ofrecer una serie de servicios durante la estancia de estas.

El hotel quiere contar con una aplicación all-in-one propia que permita tener una gestión interna de sus recursos y que también sirva para que sus clientes puedan contratar estancias para sus mascotas. Este software debe contar con un sistema de almacenamiento en la nube, debe usar tecnologías de vanguardia, debe ser ligero en cuanto a su uso y fácil de escalar. Esta aplicación, además, debe



proyectar la imagen del hotel y ofrecer una experiencia de usuario muy satisfactoria.

En esta primera fase del proyecto, se quieren cubrir los requisitos mínimos para que el hotel pueda usar ya la aplicación en su día a día, es decir, se podrán registrar los usuarios, hacer reservas, contratar servicios, listas, modificar y eliminar datos, contar con un entorno que proporcione una correcta experiencia de usuario y disponer de un sistema para el almacenamiento de datos.

### 3. Análisis de las alternativas del mercado

Nuestra aplicación pretende ser una alternativa de PMS (Property Management System) o software de gestión hostelera destinado exclusivamente a hoteles cuyos huéspedes sean animales. Actualmente existen muchas alternativas de PMS que se utilizan en los hoteles convencionales, cada uno aporta distintas ventajas, que se adaptan a las necesidades de cada negocio. A continuación, se muestran los software más destacados que servirán de inspiración para nuestro proyecto.

Cloudbeds combina herramientas de operaciones, ingresos, distribución, marketing y crecimiento, con un marketplace de integraciones de terceros para ayudar a hoteleros y anfitriones a aumentar sus ingresos y agilizar operaciones, con sistemas de almacenamiento en la nube.

Little Hotelier ofrece distintos software para la gestión de complejos hoteleros entre los que destacan el de administración y el de reservas.

Hotelguest que es un software multiplataforma serverless que ofrece sistema de identificación sin usuarios con una aplicación web, por lo que no se requiere descarga y distintos procesos automatizados.

Existen muchos más y todos cuentan con características similares, pero ninguno destinado a el tipo de negocio que tiene el cliente de este proyecto, aunque los distintos módulos o complementos que se pueden contratar se pueden solventar las necesidades.



## 4. Tecnologías empleadas

Para este proyecto, se emplearán distintas tecnologías y aplicaciones en cada uno de los aspectos que forman parte de su desarrollo y explotación. La imagen que se muestra a continuación representa el uso de las mismas y tras dicha imagen, las explicaciones y razonamientos para cada una, añadiendo otros detalles empleados en el desarrollo.



### Otras herramientas empleadas





## 4.1. Desarrollo

### 4.1.1. Back

- **MongoDB** como sistema de base de datos no relacional, por seguir la estructura de documentos JSON, ser de código abierto y la compatibilidad con JavaScript e ir en sintonía con las tecnologías punteras actuales.
- **JavaScript** como lenguaje de programación por su flexibilidad de uso tanto en el lado cliente como en servidor, por ser un lenguaje puntero con popularidad en proyectos modernos.
- **Node.js** como entorno en tiempo de ejecución para la capa del servidor, pues nos permite conectar las peticiones del cliente con la base de datos de una forma sencilla gracias a las librerías que nos permite incorporar, además de ser una tecnología moderna con mucha presencia actual. Entre las librerías empleadas destacan: mongoose (conexión con la base de datos), bcrypt (encriptado de contraseñas), passport (gestión de roles), validator (validaciones de datos), etc.
- **Express.js** como entorno de trabajo en sintonía con Node, que nos permite la creación de nuestro aplicativo web de una forma sencilla y muy potente.

### 4.1.2. Front

- **EJS** como sistema de vistas que permite embeber JavaScript en los diseños de las páginas, de manera que resulta mucho más fácil controlar la información del servidor y la experiencia de usuario a nivel visual, muy integrado en HTML y de muy fácil aprendizaje, además, permite tener un proyecto monolítico con las tecnologías punteras.
- **HTML5** como lenguaje de marcado, por ser el estándar de referencia para todos los navegadores, siendo además fácil de usar, con extensiones para VSC y rápida implementación con las plantillas de EJS.
- **CSS3** como hoja de estilo para el diseño de la interfaz por su alta aceptación, comunidad y documentación. Además por su facilidad para integrarse en el HTML y contar con librerías como Bootstrap. **Bootstrap** como framework para el diseño de vistas de la aplicación, por su fácil



integración, uso y comunidad, que respalda y facilita la integración de componentes y estilos accesibles.

## 4.2. Despliegue

- **Railway** como servidor de despliegue del proyecto, por su facilidad de uso, compatibilidad con las tecnologías empleadas y la disponibilidad de 500 horas de uso gratuitas.
- **MongoDB Atlas** como servicio de base de datos multi-cloud, que permite desplegar y manejar bases de datos de una forma sencilla y de última tecnología.

## 4.3. Útiles

### 4.3.1. Construcción

- **Sistema operativo Windows 10**, que cuenta con una alta compatibilidad con las herramientas que se emplean, gran aceptación y respaldo, y facilidad de uso.
- **Navegador Chrome**, por su facilidad de uso y las extensiones que permite incorporar, siendo, además, el navegador más utilizado tanto en ordenadores como en móviles.
- **Visual Studio Code** como editor de código por su ligereza, facilidad de uso y las extensiones que ayudan enormemente al desarrollo como son: GitLens, JavaScript Code Snippets, Tabnine, Auto Close Tag, Auto Rename Tag, Prettier y Color Highlight.

### 4.3.2. Control

- **GitHub** para alojar el proyecto y tener un control de versiones, y acceso al mismo desde Internet.
- **Trello** como herramienta para administrar el desarrollo ágil del proyecto.

### 4.3.3. Diseño

- **Figma** como editor gráfico para el diseño visual de las páginas tanto para el wireframe como el mockup, pues facilita el diseño por sus distintas funcionalidades como la creación de componentes, grupos y las extensiones que se le pueden añadir como Iconify, el plugin de los iconos.



#### 4.3.4. Pruebas

- **Postman** como cliente para usar las API antes de relacionarlas con vistas, permite organizar las distintas peticiones y tratar JSON, es fácil de instalar y usar.

#### 4.3.5. Otros

- **Google Documents** para la creación del manual de despliegue y la memoria del proyecto. Esta herramienta permite tener toda la información almacenada en la nube de manera inmediata, nos aporta un sistema de indexado automático, personalización de estilos, realización de comentarios y la conversión a PDF, entre muchas otras opciones, además de un sistema seguro de disponibilidad, acceso y facilidad de uso.
- **Google fonts** como directorio de fuentes para la web, por ser gratuito y ser propiedad de una empresa con reconocimiento mundial.

## 5. Diseño de la aplicación

Para poner en marcha el proyecto es necesario definir una serie de elementos previamente.

### 5.1. Funcionalidades

En primer lugar, se deben tener en cuenta qué requisitos funcionales debe cumplir el sistema. Estos requisitos están identificados y priorizados, por lo que el desarrollo se realiza, preferiblemente, por orden de prioridad.

| ID   | Prioridad | Título   | Descripción  |
|------|-----------|----------|--|
| RF-1 | 100       | Acceso   | Un usuario anónimo debe poder acceder a la página de inicio, ver los servicios, las instalaciones, la ubicación de la empresa e interactuar con el sistema |
| RF-2 | 100       | Registro | Un usuario anónimo se debe poder   |



|       |     |                     |  |
|-------|-----|---------------------|--|
|       |     |                     | registrar  |
| RF-3  | 100 | Inicio sesión       | Un usuario registrado con cualquier rol debe poder iniciar sesión  |
| RF-4  | 90  | Cierre sesión       | Un usuario registrado con cualquier rol debe poder cerrar sus sesión sin perder sus datos  |
| RF-5  | 90  | Editar perfil       | Un usuario registrado con cualquier rol debe poder modificar algunos datos de su perfil  |
| RF-6  | 80  | Resetear contraseña | Un usuario registrado con cualquier rol debe poder resetear su contraseña en caso de olvido  |
| RF-7  | 50  | Darse de baja       | Un usuario registrado con rol cliente debe poder solicitar darse de baja, perdiendo el acceso al servicio  |
| RF-8  | 100 | Añadir mascota      | Un usuario registrado con rol cliente debe poder añadir mascotas a su perfil   |
| RF-9  | 100 | Editar mascota      | Un usuario registrado con rol cliente debe poder modificar algunos datos de sus mascotas   |
| RF-10 | 100 | Eliminar mascota    | Un usuario registrado con rol cliente debe poder solicitar la eliminación de una mascota de su espacio (No debe ser una eliminación real en la bd) |
| RF-11 | 100 | Crear reserva       | Un usuario registrado con rol cliente debe poder realizar la reserva de una habitación   |
| RF-12 | 100 | Editar reserva      | Un usuario registrado con rol cliente debe poder editar una reserva que tenga activa   |
| RF-13 | 100 | Eliminar reserva    | Un usuario registrado con rol cliente debe poder eliminar una reserva que tenga activa   |
| RF-14 | 80  | Listar reservas     | Un usuario registrado con rol cliente debe poder ver todas sus reservas, activas o no  |
| RF-15 | 80  | Listar mascotas     | Un usuario registrado con rol cliente  |



|       |     |   |  |
|-------|-----|---|--|
|       |     |   | debe poder ver todas sus mascotas  |
| RF-16 | 60  | Contratar servicio                      | Un usuario registrado con rol cliente debe poder añadir servicios en la reserva  |
| RF-17 | 50  | Editar contrato servicio                | Un usuario registrado con rol cliente debe poder editar los servicios contratados en una reserva                           |
| RF-18 | 40  | Eliminar contrato servicio              | Un usuario registrado con rol cliente debe poder eliminar los servicios contratados  |
| RF-49 | 100 | Listar usuarios                         | Un usuario registrado con rol administrador puede ver todos los usuarios del sistema                                       |
| RF-20 | 90  | Listar clientes                         | Un usuario registrado con rol gerente o recepcionista puede listar los usuarios con rol cliente                            |
| RF-21 | 50  | Ver ganancias                           | Un usuario registrado con rol gerente puede visualizar una gráfica con las ganancias mensuales obtenidas por las estancias |
| RF-22 | 30  | Ver estadística de servicios            | Un usuario registrado con rol gerente puede visualizar una gráfica con los servicios más demandados                        |
| RF-23 | 30  | Ver estadística de tipo de habitaciones | Un usuario registrado con rol gerente puede visualizar una gráfica con el tipo de habitaciones más demandado               |
| RF-24 | 50  | Listar empleados                        | Un usuario registrado con rol gerente puede visualizar las fichas de los empleados   |
| RF-25 | 10  | Añadir puntuación                       | Un usuario registrado con rol cliente puede añadir una puntuación de su estancia   |
| RF-26 | 70  | Buscar un usuario                       | Un usuario registrado con rol staff o admin debe poder busca a un usuario por su username y ver todos sus datos            |
| RF-27 | 70  | Buscar una mascota                      | Un usuario registrado con rol staff o admin debe poder busca a una   |



|       |    |                            |  |
|-------|----|----------------------------|--|
|       |    |                            | mascota por su id y ver todos sus datos  |
| RF-28 | 70 | Buscar una reserva         | Un usuario registrado con rol staff o admin debe poder busca a una mascota por su id y ver todos sus datos                     |
| RF-29 | 50 | Generar PDF con la reserva | Un usuario autenticado debe poder tener acceso a un documento PDF que se genere con los datos de la reserva asociada           |
| RF-30 | 40 | Subir foto de perfil       | Un usuario autenticado debe poder cambiar su foto de perfil pudiendo subirla desde su ordenador                                |
| RF-31 | 40 | Subir foto de mascota      | Un usuario autenticado debe poder cambiar la foto de su mascota pudiendo subirla desde su ordenador                            |
| RF-32 | 10 | Copiar id en portapapeles  | En todos los listados de la aplicación que salen id o username se debe poder copiar el dato en el portapapeles del dispositivo |

## 5.2. Diseño de la base de datos

Por otra parte, en cualquier proyecto es fundamental contar con un sistema que ayude a gestionar los datos contenidos, para ello, es fundamental disponer de un modelo de datos y la relación que existe entre las diferentes colecciones. Tal y como se ha mencionado en el apartado del stack tecnológico, para este proyecto se utilizará una base de datos no relacional, MongoDB, que nos permitirá tener una gran flexibilidad en cuanto a la información contenida.

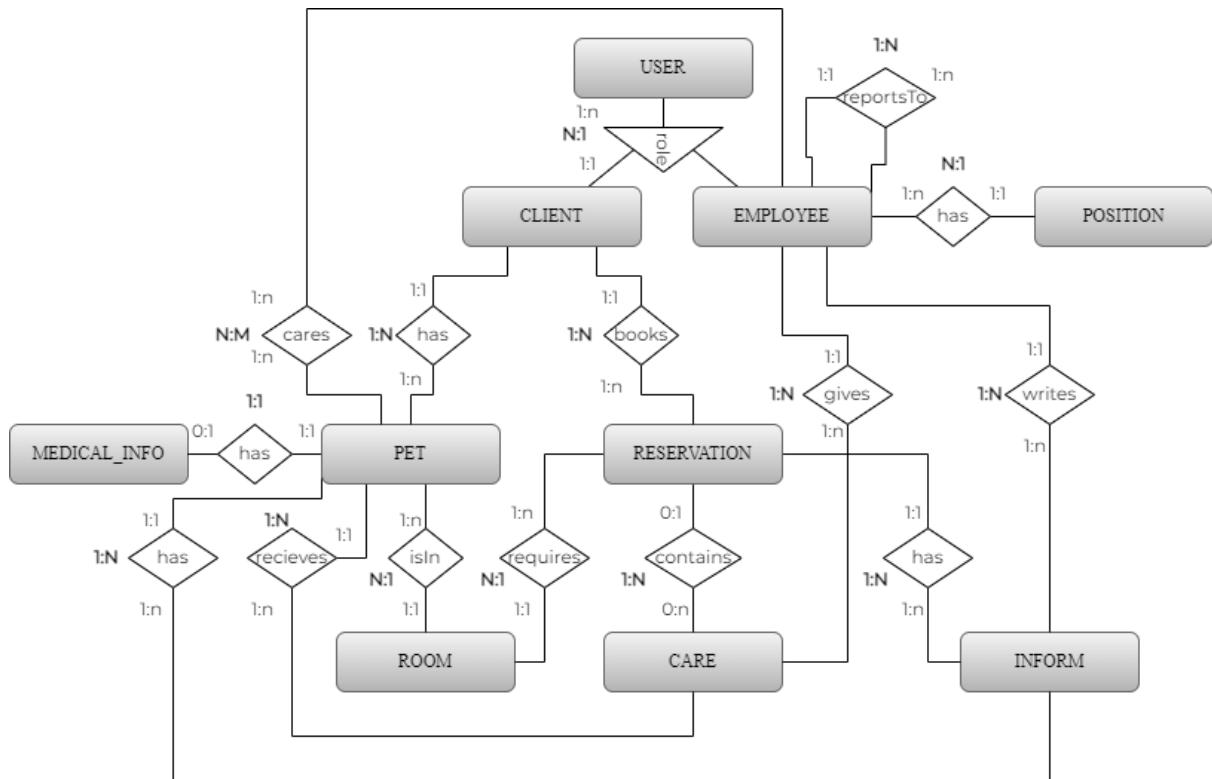
### 5.2.1. Esquema

En el siguiente gráfico se muestra la relación de las distintas entidades que componen la base de datos del proyecto.

Encontramos la entidad User, de la nacen los tipos Client y Employee, pues tienen algunos atributos distintos. Entre los empleados encontramos el responsable de cada trabajador y su puesto en la empresa. Algunos empleados se encargan de las mascotas, presentan informes y prestan servicios. Los clientes, por otro lado,



tienen mascotas y realizan reservas para las mismas. Cada mascota tiene una información médica y se aloja en una habitación, la cual es asignada en la reserva. Si observamos, las entidades con más impacto en nuestra base de datos son User, Pet y Reservation.



### 5.2.2. Modelado de datos

En este proyecto contamos con colecciones que se estructuran de la siguiente manera:

| COLECCIÓN: USER (USUARIOS DEL SISTEMA) |   |   |
|--|---|---|
| Atributo                               | Características   | Descripción   |
| username                               | - tipo cadena<br>- obligatorio<br>- ordenación por elemento único | Identificador de tipo cadena con un nombre elegido por el usuario, el cual no puede estar registrado en la base de datos, es de ingreso |



|          |   |  |
|----------|---|--|
|          |   | obligatorio y se le crea un index para una búsqueda más eficiente  |
| name     | - tipo cadena<br>- obligatorio                                    | Nombre del usuario como campo obligatorio de tipo cadena   |
| surnames | - tipo cadena<br>- obligatorio                                    | Apellidos del usuario como campo obligatorio de tipo cadena  |
| password | - tipo cadena<br>- obligatorio                                    | Contraseña del usuario que le permite el acceso al sistema. Esta cumplirá con unas validaciones y se almacenará encriptada. Este campo es obligatorio y de tipo cadena |
| idnumber | - tipo cadena<br>- obligatorio<br>- ordenación por elemento único | Número de identidad del usuario (dni, nie o pasaporte) como campo obligatorio, único e indexado para facilitar la búsqueda y es de tipo cadena                         |
| email    | - tipo cadena<br>- obligatorio<br>- ordenación por elemento único | Correo electrónico del usuario como campo obligatorio, único e indexado para facilitar la búsqueda y de tipo cadena  |
| phone    | - tipo cadena<br>- obligatorio                                    | Número de teléfono del usuario como campo obligatorio y de tipo cadena   |
| address  | - tipo cadena<br>- obligatorio                                    | Dirección del usuario como campo obligatorio y de tipo cadena  |
| birthday | - tipo fecha<br>- obligatorio                                     | Fecha de nacimiento del usuario como campo obligatorio y de tipo   |



|               |  |  |
|---------------|--|--|
|               |  | fecha. Se validará si el usuario es mayor de edad  |
| register_date | <ul style="list-style-type: none"><li>- tipo fecha</li><li>- obligatorio</li><li>- valor por defecto, la hora actual</li></ul> | Fecha en la que se realiza el registro del usuario (se crea), que tiene por defecto la fecha en la que se realiza, siendo de almacenamiento obligatorio y de tipo fecha, no introducible, solo legible.  |
| avatar        | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- valor por defecto, la ruta de una imagen</li></ul>               | Url de la foto de perfil del usuario, que será de tipo cadena y que tendrá un valor por defecto en caso de no introducir ninguno   |
| status        | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- tipo enumerado</li><li>- valor por defecto, activo</li></ul>     | Estado del usuario para el sistema, de manera que si se encuentra activo podrá realizar todas las acciones correspondientes y si está inactivo, solo queda registrado en la base de datos, pero no tiene acceso a nada. Este campo es de tipo cadena y sólo admite dos valores, siendo 'active' el valor por defecto |
| role          | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- tipo enumerado</li><li>- valor por defecto, cliente</li></ul>    | Rol del usuario para el sistema, solo se podrán crear 'client' desde el registro "anónimo", siendo los roles admin y staff los únicos con capacidad de asignar el rol de empleado. Este campo es de tipo cadena y sólo admite tres valores: 'client', 'staff' y 'admin', siendo 'client' el valor por defecto        |



|      |   |  |
|------|---|--|
| pets | <ul style="list-style-type: none"><li>- tipo array</li><li>- almacén de ObjectId de Mongo con referencia a Pet</li><li>- valor por defecto, array vacío</li></ul> | Arreglo de mascotas del usuario, en el que se referencian los identificadores de las mismas, por defecto se carga en vacío |
|------|---|--|

**Documento embebido - position: almacena los datos del puesto ejercido**

|            |   |  |
|------------|---|--|
| start_date | <ul style="list-style-type: none"><li>- tipo fecha</li><li>- valor por defecto, fecha actual</li></ul>                    | Fecha de incorporación en la empresa, que por defecto es la fecha en la que se crea el usuario   |
| end_date   | <ul style="list-style-type: none"><li>- tipo fecha</li></ul>  | Fecha de baja en la empresa, que se rellenará cuando el empleado se dé de baja   |
| studies    | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>   | Cadena con la descripción de los estudios que tenga el empleado  |
| title      | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- tipo enumerado</li><li>- valor por defecto, vacío</li></ul> | Permite saber el puesto que desempeña el usuario en la empresa. Temporalmente los valores son 'employee' para todos los empleados en general y 'manager' para los gerentes |
| salary     | <ul style="list-style-type: none"><li>- tipo numérico</li><li>- valor por defecto, 0</li></ul>                            | Almacena lo que el miembro del staff cobra   |
| reportsTo  | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- valor por defecto, vacío</li></ul>                          | Almacena el nombre del usuario "jefe" para el empleado   |

**COLECCIÓN: PET (MASCOTAS DE LOS USUARIOS)**

| Atributo      | Características   | Descripción   |
|---------------|---|---|
| _id           | - tipo ObjectId de Mongo<br>- autogenerado  | Identificador autogenerado por Mongo único y eficiente, no editable, ni introducible, sólo legible  |
| register_date | - tipo fecha<br>- obligatorio<br>- valor por defecto, la fecha actual                 | Fecha en la que se registra a la mascota que, por defecto, será la actual y es de tipo obligatoria. No es editable ni introducible, sólo legible                                      |
| name          | - tipo cadena<br>- obligatorio  | Nombre de la mascota obligatorio  |
| birthday      | - tipo fecha<br>- obligatorio   | Fecha de nacimiento de la mascota obligatoria   |
| chip          | - tipo cadena   | Código del microchip que tenga integrado, si lo tiene   |
| specie        | - tipo cadena<br>- tipo enumerado<br>- tipo obligatorio<br>- valor por defecto, perro | Tipo de mascota, que sólo podrá ser de alguno de los tipos que admite el hotel: perro, gato, roedor, ave, pez y otro, siendo el perro el valor por defecto. Este campo es obligatorio |
| breed         | - tipo cadena   | Detalle de la raza de la mascota  |
| size          | - tipo cadena<br>- tipo enumerado<br>- valor por defecto, mediano                     | Tamaño de la mascota, que estará en un rango: pequeño, mediano o grande, siendo por defecto mediano el valor asignado   |
| color         | - tipo cadena   | Color de la mascota   |



|              |  |  |
|--------------|--|--|
| friendly     | <ul style="list-style-type: none"><li>- tipo boolean</li><li>- tipo obligatorio</li><li>- valor por defecto, verdadero</li></ul> | Campo boolean para indicar si la mascota es sociable con otras mascotas de su misma especie u otras  |
| avatar       | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- valor por defecto, la ruta de una imagen</li></ul>                 | Url de la foto de la mascota, que será de tipo cadena y que tendrá un valor por defecto en caso de no introducir ninguno                                   |
| comments     | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>  | Campo de tipo cadena para introducir comentarios sobre la mascota  |
| status       | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- tipo enumerado</li><li>- valor por defecto, activo</li></ul>       | Campo para indicar el estado de la mascota, si está activa o no, en el caso de no estar activo, no se mostrará pero seguirá almacenado en la base de datos |
| medical_info | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>  | Campo de tipo cadena que contiene información médica de la mascota, como enfermedades, el tipo y el tratamiento que tenga                                  |
| caregiver    | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>  | Campo que referencia al usuario al que pertenece esta mascota, por su username   |

**Documento embebido - inform: almacena los informes de cada mascota**



|             |   |  |
|-------------|---|--|
| timestamp   | <ul style="list-style-type: none"><li>- tipo fecha</li><li>- valor por defecto, la fecha actual</li></ul> | Permite recoger el momento de generación del informe, no es un campo introducible, sólo legible.         |
| author      | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>   | Almacena el nombre de usuario (su username) del miembro del staff que lo genera, capturado por la sesión |
| shift       | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- tipo enumerado</li></ul>                    | Indica el turno al que pertenece el informe, teniendo tres posibles valores: mañana, tarde y noche       |
| description | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>   | Almacena toda la información que se requiere en el informe   |

| COLECCIÓN: RESERVATION (RESERVAS DE LOS USUARIOS) |   |  |
|---|---|--|
| Atributo  | Características   | Descripción  |
| _id   | <ul style="list-style-type: none"><li>- tipo ObjectId de Mongo</li><li>- autogenerado</li></ul> | Identificador autogenerado por Mongo único y eficiente, no editable, ni introducible, sólo legible                 |
| checkin   | <ul style="list-style-type: none"><li>- tipo fecha</li><li>- obligatorio</li></ul>              | Fecha de entrada obligatoria, se evalúa que la fecha nunca pueda ser anterior que el momento actual                |
| checkout  | <ul style="list-style-type: none"><li>- tipo fecha</li><li>- obligatorio</li></ul>              | Fecha de salida obligatoria, se evalúa que sea, como mínimo, el día posterior a la fecha introducida en el checkin |



|              |  |  |
|--------------|--|--|
| timestamp    | <ul style="list-style-type: none"><li>- tipo fecha</li><li>- valor por defecto, la fecha actual</li></ul>  | Fecha de realización de la reserva, que por defecto es la que se toma en el momento en el que se realiza la reserva. No introducible ni editable, sólo legible   |
| client       | <ul style="list-style-type: none"><li>- tipo cadena</li><li>- obligatorio</li></ul>  | Campo que referencia al usuario titular de la reserva por su username. Si el usuario que realiza la reserva es el propio cliente, captura el username de la sesión y es un valor sólo legible, si no lo es, es un campo editable |
| pets         | <ul style="list-style-type: none"><li>- tipo array</li><li>- almacén de ObjectId de Mongo referenciando a la mascota</li><li>- obligatorio</li></ul> | Mascotas implicadas en la reserva, que serán las que se alojen. Se almacena en este array de objetos los identificadores de dichas mascotas, debe haber al menos una mascota para que se pueda realizar la reserva               |
| petsitter    | <ul style="list-style-type: none"><li>- tipo array</li><li>- almacén de cadenas con los nombre de usuarios</li></ul>                                 | Arreglo de usuarios de tipo staff (por su username) que están a cargo de los cuidados de la mascota  |
| observations | <ul style="list-style-type: none"><li>- tipo cadena</li></ul>  | Campo de tipo cadena para introducir las observaciones necesarias sobre la reserva; este campo es editable por el usuario cliente  |



|        |  |   |
|--------|--|---|
| cares  | - tipo cadena  | Campo de tipo cadena que permite indicar los cuidados extra que se desean contratar a la hora de hacer la reserva                                 |
| price  | - tipo numérico  | Campo para almacenar el precio que debe pagar el usuario por la estancia completa   |
| status | - tipo cadena<br>- tipo enumerado<br>- obligatorio<br>- valor por defecto, pendiente | Campo de tipo cadena que admite los valores "active", por defecto, e "inactive" que se cambia cuando el usuario se quiere dar de baja del sistema |

**Documento embebido - room: almacena los datos de la habitación**

|          |  |  |
|----------|--|--|
| type     | - tipo cadena<br>- tipo enumerado<br>- obligatorio<br>- valor por defecto, suite | Permite almacenar el tipo de habitación en la que el cliente quiere alojar a su mascota  |
| assigned | - tipo cadena  | Almacena una cadena que incluye el id propio de la habitación donde se alojará la mascota. Este campo sólo se puede llenar desde un perfil staff o admin |
| details  | - tipo cadena  | Almacena una cadena con cualquier detalle que se quiera proporcionar de la habitación  |



## 5.3. Vistas

### 5.3.1. Wireframe

| Vistas | Usos  |
|--------|---|
|        | Página de inicio (para todos los usuarios)<br><br>Secciones: <ul style="list-style-type: none"><li>- Menú superior</li><li>- Acción directa a reserva</li><li>- Presentación</li><li>- Habitaciones</li><li>- Servicios</li><li>- Pie de página</li></ul> |
|        | Página de registro (para todos los usuarios)<br><br>Página de editar perfil (para los usuarios logados)<br><br>Página de crear o editar una reserva (para los usuarios logados)   |
|        | Pop-up para realizar el login (para todos los usuarios)   |
|        | Página de resetear la contraseña (para todos los usuarios)  |
|        | Página de inicio para un usuario logado de tipo cliente   |



|  |   |
|--|---|
|  | Página de inicio para un usuario logado de tipo empleado  |
|  | Página de inicio para un usuario logado de tipo gerente   |
|  | Página de inicio para un usuario logado de tipo administrador   |
|  | <p>Página para ver listado de reservas (para los usuarios logados - clientes, empleados, gerentes y admin)</p> <p>Página para ver listado de mascotas (para los usuarios logados - clientes, empleados, gerentes y admin)</p> <p>Página para ver listado de usuario (para los usuarios logados - gerentes y admin)</p> <p>Página para ver listado de informes (para los usuarios logados - empleados, gerentes y admin)</p> |
|  | <p>Página para ver los informes (para los usuarios logados - empleados, gerentes y admin)</p> <p>Página para crear los informes (para los usuarios logados - empleados, gerentes y admin)</p> <p>Página para editar los informes (para los usuarios logados - empleados, gerentes y admin)</p>  |



### 5.3.2. Mockup

## Vistas

**PETLAND RESORT**  
CANTERA DE LA ALMENDRA

**Quiénes somos**

PETLAND RESORT nació de la mano de amantes de los animales con la visión de ofrecer la mejor experiencia para el cuidado y bienestar integral de sus mascotas en un entorno acogedor y seguro.

Oferemos distintas tipos de instalaciones según las preferencias o necesidades de cada animal y sus necesidades específicas de su entorno de vida y crecimiento.

Nuestro equipo está formado por profesionales altamente cualificados en su campo y el desarrollo y crecimiento de sus mascotas es su principal objetivo.

[Aprende más...](#)

**Nuestras habitaciones**

Suite Individual [Reserva](#)

Suite Familiar [Reserva](#)

Deluxe sans individuel [Reserva](#)

Deluxe suite familiar [Reserva](#)

**Nuestros servicios**

Terapias [Reserva](#)

Educa y Juega [Reserva](#)

Entrenamiento [Reserva](#)

Servicio personalizado [Reserva](#)

**Nuestros datos**

Validez de los nuestros clientes:

+3.000 Mascotas atendidas

+150 Eventos celebrados

**PETLAND RESORT**  
CANTERA DE LA ALMENDRA

"Una experiencia de lujo para seres de lujo"

**Encuéntranos**

Carrilera Majana, S/N, 41807 Espartinas, Sevilla  
+34 958 123 456  
info@petlandresort.com

**Síguenos**

[Facebook](#) [Instagram](#)

Todos los derechos reservados para Petland Resort 2022

Página de inicio, donde llegan todos los usuarios



# Regístrate

NOMBRE

Apellidos

DNI/NIE

Correo electrónico

Número de teléfono

Dirección

Nombre de usuario

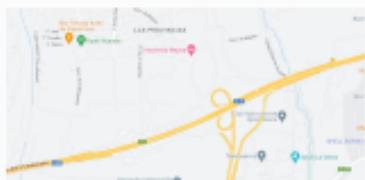
Contraseña de acceso

Acepto las políticas de privacidad

[Cancelar](#) [Registrar](#)



Encuéntranos



Camino Mojina, 5/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

Síguenos

[!\[\]\(f159d9550cba63c9a191c936a406412a\_img.jpg\)](#) [!\[\]\(1aaa31036622bbfcd5b363a81174d9a7\_img.jpg\)](#)

Todos los derechos reservados para Petland Resort 2023

Página para registrarse como cliente



Iniciar sesión

Nombre de usuario

Contraseña de acceso

Recordarme

**Entrar**

He olvidado mi contraseña

¿Aún no estás registrado? [Regístrate aquí](#)

Encuéntranos

Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

— Síguenos —

f o

Todos los derechos reservados para Petland Resort 2023

Pop-up para iniciar sesión



PETLAND RESORT

Andrea Vieira

PETLAND RESORT

Inicio Reservas Habitaciones Servicios Nosotros Login

Restaurar contraseña

Correo electrónico

Nueva contraseña de acceso

Enviar

Encuéntranos

Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

Todos los derechos reservados para Petland Resort 2023

Página para el restablecimiento de la contraseña

Síguenos

f o



PETLAND RESORT

Inicio Reservas Habitaciones Servicios Nosotros Ana

Hola, Ana

Aquí te mostramos tu espacio personal donde puedes gestionar tu perfil, tus mascotas y tus reservas

Editar perfil

Mis mascotas Mis reservas Mis servicios

### Próxima reserva

Del 03/04/2023 al 06/04/2023 (3 días)

Referencia de la reserva: 507f91e810c19729de860ea

Nº de habitaciones: 1

Nº de ocupantes: 1

Observaciones: lleva su peluche favorito

Servicios: peluquería y spa

Editar reserva Cancelar reserva

### Mis mascotas

**Nombre: Rex**

Identificador: 507f91e810c19729de860ea

Raza: Braco de Weimar

Editar mascota Listar mascotas Añadir mascota

### Encuéntranos

Camino Mejina, s/n, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

### Síguenos

Todos los derechos reservados para Petland Resort 2020

Página de inicio al cliente tras el login

26



PETLAND RESORT

Inicio Reservas Habitaciones Servicios Nosotros Cristina

Hola, Cristina

Aquí te mostramos tu espacio personal donde puedes gestionar tu perfil y consultar tus datos

[Editar perfil](#)

## Panel de control

Buscar reserva

Identificador de la reserva

Buscar mascota

Identificador de la mascota

Buscar cliente

Identificador del cliente

Buscar servicio

Identificador del servicio

Acciones

Mi calendario

Informes generados

Ver perfil personal

Datos profesionales

PETLAND RESORT

"Una experiencia de lujo para seres de lujo"

Encuéntranos

Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

Síguenos

Todos los derechos reservados para Petland Resort 2023

Página de inicio a empleados que no son gerentes tras el login



Hola, Victoria

Aquí te mostramos tu espacio personal donde puedes gestionar su perfil y consultar tus datos

[Editar perfil](#)

[Lista reservas](#) [Lista servicios](#) [Lista habitaciones](#) [Lista mascotas](#) [Lista clientes](#)

### Panel de control

Miembros del personal

- Gerente
- Colaborador
- Recepcionista
- Todos

Gastos de personal

Perspectiva general de reservas

Perspectiva general de ganancias

Perspectiva de mascotas

Perspectiva de servicios

Perspectiva de habitaciones

PETLAND RESORT

"Una experiencia de lujo para seres de lujo"

Encuéntranos

camino Mejía, s/n, 46822 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

Síguenos

[Facebook](#) [Instagram](#)

Todos los derechos reservados para Petland Resort 2023

Página de inicio para los empleados gerentes tras el login

28



Hola, Andrea

Este es tu espacio como administradora de la aplicación

## Panel de administración

**Centro**

Habitaciones: [Añadir habitación](#) [Listar habitaciones](#) [Eliminar habitación](#)

Servicios: [Añadir servicio](#) [Listar servicios](#) [Eliminar servicio](#)

Gerencia: [Ganancias](#) [Gastos](#)

**Usuarios**

Empleados: [Listar todos los usuarios](#) [Añadir empleado](#) [Listar empleados](#) [Eliminar empleado](#)

Clientes: [Añadir cliente](#) [Listar clientes](#) [Eliminar cliente](#)

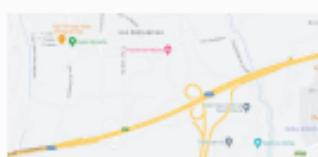
**Mascotas**

[Añadir mascota](#) [Listar mascota](#) [Eliminar mascota](#)

**Reservas**

[Añadir reserva](#) [Listar reservas](#) [Eliminar reserva](#)

## Encuéntranos

  
Camino Mejina, 5N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

## Síguenos

Todos los derechos reservados para Petland Resort 2023

Página de inicio del administrador de la aplicación tras el login





PETLAND RESORT

Inicio Reservas Habitaciones Servicios Nosotros  Ana

## Nueva reserva

Fecha de entrada  Fecha de salida

Tipo de habitación  Número de huéspedes

Observaciones

Acepto las políticas de privacidad

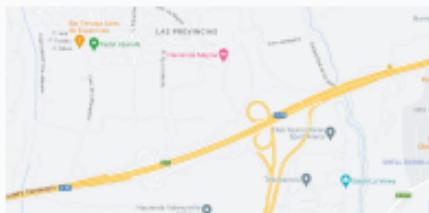


## Encuéntranos



PETLAND RESORT

"Una experiencia de lujo para seres de lujo"



Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

### Síguenos

Todos los derechos reservados para Petland Resort 2023

Página para iniciar una reserva



## Detalles de la reserva

**Huésped 1:**

Selección mascota:

Servicios:

Masaje    Peluquería    Spa  
 Agility    Adiestramiento    Paseo extra

**Huésped 2:**

Selección mascota:

Servicios:

Masaje    Peluquería    Spa  
 Agility    Adiestramiento    Paseo extra

**Huésped 3:**

Selección mascota:

Servicios:

Masaje    Peluquería    Spa  
 Agility    Adiestramiento    Paseo extra

**Identificador de la reserva:**  
502f91e800c9729de860ea

**Fecha de entrada:** 01/03/2023   **Fecha de salida:** 04/03/2023

**Tipo de habitación:** Suite familiar   **Nº de huéspedes:** 3

**Observaciones:**  
Necesito recogerlos por la noche

**Importe estimado:** 350,50€   **Método de pago:**  Seleccionar

Recuerde revisar los datos de sus mascotas y sus datos de contacto, pues contactaremos con usted para confirmar la reserva.

Gracias por elegirnos.  
Nos vemos pronto.

**CANCELAR**   **CONFIRMAR**

## Encuéntranos



Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456  
info@petlandresort.com

**Síguenos**

Todos los derechos reservados para Petland Resort 2023

Página para añadir datos a la reserva





PETLAND RESORT  
95 95 95 95

Inicio Reservas Habitaciones Servicios Nosotros  Ana

## Detalles de la reserva

**Huesped 1:**  
Sección mascota

Servicios:  
 Masaje  Peluq  
 Agility  Adiest

**Huesped 2:**  
Sección mascota

Servicios:  
 Masaje  Peluq  
 Agility  Adiest

**Huesped 3:**  
Sección mascota

Servicios:  
 Masaje  Peluquería  Spa  
 Agility  Adiestamiento  Paseo extra

Identificador de la reserva:  
507791e8f0c9729dc660ea

Fecha de salida:  
04/03/2023

Nº de huéspedes:  
3

Nº noche:  
1 noche

Método de pago:  
Presta cash

Gracias por confirmar su reserva

Puede consultar su reserva en cualquier momento  
Puede modificar la reserva hasta 48 horas antes

[Ir a inicio](#) [Ver reservas](#)

Recuerde revisar los datos de su mascota y sus datos de contacto, pues comprobaremos con usted para confirmar la reserva.

Gracias por elegirnos  
Nos vemos pronto

[Cancelar](#) [Confirmar](#)



PETLAND RESORT  
95 95 95 95

"Una experiencia de lujo para seres de lujo"

## Encuéntranos



Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456  
info@petlandresort.com

— Síguenos —

Todos los derechos reservados para Petland Resort 2023

Pop-up para la confirmación de los datos



PETLAND RESORT

Inicio Reservas Habitaciones Servicios Nosotros  24h

## Detalles de la reserva

**Huesped 1:**

Reservación realizada

Servicios:

Masaje  Peluquería

Agility  Adiestramiento

**Huesped 2:**

Reservación realizada

Servicios:

Masaje  Peluquería

Agility  Adiestramiento

**Huesped 3:**

Reservación realizada

Servicios:

Masaje  Peluquería  Spa

Agility  Adiestramiento  Paseo extra

**Identificador de la reserva:**

507ff91e810c9f729de860ea

**Fecha de salida:**

04/03/2023

**Nº de huéspedes:**

3

**Hora:**

a noche

**Método de pago:**

Transferencia

**¿Desea eliminar su reserva?**

Una vez cancelada la reserva no se puede recuperar. Si lo desea, puede modificarla o realizar una nueva

**Cancelar** **Eliminar**

Recuerde revisar los datos de sus mascotas y sus datos de contacto, pues contactaremos con usted para confirmar la reserva.

Gracias por elegirnos.  
Nos vemos pronto

**Cancelar** **Confirmar**

## Encuéntranos



Camino Mejina, s/n, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

### Síguenos

f o

Todos los derechos reservados para Petland Resort 2023

Pop-up para la cancelación de una reserva



## Listado de reservas

| Identificador de la reserva | Fecha de entrada | Fecha de salida | Nº de huéspedes | Tipo de habitación | Opciones  |
|-----------------------------|------------------|-----------------|-----------------|--------------------|---|
| 507f191e810cf9729de860ea    | 01/03/2023       | 04/03/2023      | 3               | Suite familiar     | <a href="#">Editar reserva</a> <a href="#">Cancelar reserva</a>                     |
| 507f191e810cf9729de860ea    | 23/04/2023       | 26/04/2023      | 3               | Suite familiar     | <a href="#">Editar reserva</a> <a href="#">Cancelar reserva</a>                     |
| 507f191e810cf9729de860ea    | 30/12/2022       | 02/01/2023      | 3               | Suite familiar     |   |
| 507f191e810cf9729de860ea    | 23/12/2022       | 26/12/2022      | 3               | Suite familiar     |  |

## Encuéntranos

Camino Mejina, s/n, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

**Síguenos**

[Facebook](#) [Instagram](#)

Todos los derechos reservados para Petland Resort 2023

Página para listar las reservas



## Listado de mascotas

Buscar mascota  Identificador  Ordenar por tipo  Ordenar alfabéticamente

|  |  |
|--|--|
| <b>Nombre:</b> Rex<br>Identificador de la mascota: 507f191e810c19729de860ea <input type="button" value=""/> Tipo: perro<br>Raza: Braco de Weimar<br>Fecha de nacimiento: 21/07/2013 (9 años) |  |
| <b>Nombre:</b> Nala<br>Identificador de la mascota: 987f191e810c19729de860ea <input type="button" value=""/> Tipo: perro<br>Raza: Pastor alemán<br>Fecha de nacimiento: 24/03/2019 (3 años)  |  |
| <b>Nombre:</b> Salem<br>Identificador de la mascota: 967f191e810c19729de860ba <input type="button" value=""/> Tipo: gato<br>Raza: Korat<br>Fecha de nacimiento: 01/01/2019 (4 años)          |  |

## Encuéñtranos

**PETLAND RESORT**

“Una experiencia de lujo para seres de lujo”

Camino Mejina, 5/N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

— Síguenos —

Todos los derechos reservados para Petland Resort 2023

Página para listar las mascotas



Añadir mascota

Nombre  Fecha de nacimiento

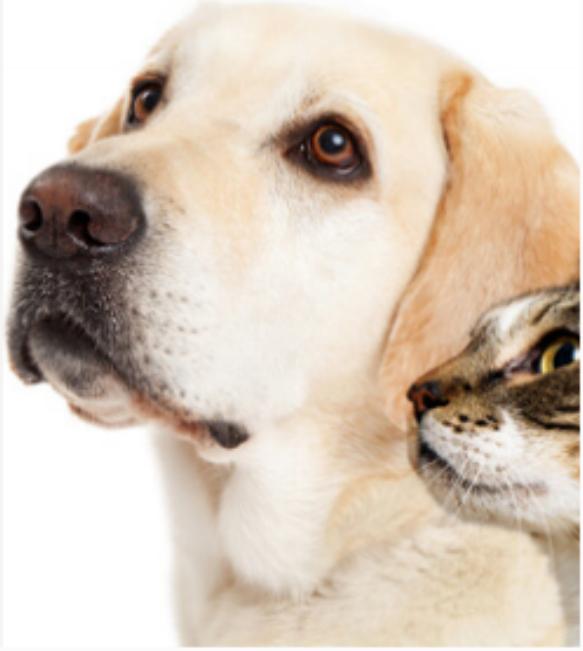
Tipo de mascota  Raza

Tamaño  Color

Chip  Número del chip integrado  Sociable

Datos médicos  Indicar enfermedad y tratamiento

Comentarios  Detalles sobre cosas que le gustan o que no le gustan...



 PETLAND RESORT

—  —

“Una experiencia de lujo para seres de lujo”

Encuéntranos



Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

— Síguenos —

Todos los derechos reservados para Petland Resort 2023

Página para añadir mascota



**Mi mascota**

|  |                                   |
|--|-----------------------------------|
| Nombre   | Fecha de nacimiento               |
| Rex  | 29/07/2013                        |
| Tipo de mascota                                      | Raza                              |
| Perra  | Braco de Weimar                   |
| Tamaño   | Color                             |
| Mediano  | Gris                              |
| Número de identificación                             |                                   |
| 507791981010729de890aa                               |                                   |
| Chip   | <input type="checkbox"/> Sociable |
| Datos médicos  |                                   |
| Oídos delicados, necesita limpieza diaria            |                                   |
| Comentarios  |                                   |
| Le gustan mucho jugar y el agua. No come pienso seco |                                   |

**Ver informes**

**Inactivar** **Activar**

  
**Cambiar foto de perfil**

**Encuéntranos**



Camino Mejina, S/N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

**Síguenos**

[f](#) [o](#)

Todos los derechos reservados para Petland Resort 2023

Página para ver y/o editar una mascota



The screenshot shows a form titled "Nuevo informe" (New Report) on the Petland Resort website. The form fields include:

- Personas a cargo: Identificador de empleado (Employee ID) and Seleccionar mascota (Select pet).
- Número de habitación: N° de habitación (Room number) and Seleccionar turno (Select shift).
- Fecha: 04/04/2023.
- Descripción: Detalles sobre estado, comportamiento y revisión (Details about state, behavior and review).

Buttons at the bottom are "Cancelar" (Cancel) and "Añadir" (Add).



## Encuéntranos



Camino Mejina, S/N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

**Síguenos**

f o

Todos los derechos reservados para Petland Resort 2023

Página para añadir informe





PETLAND RESORT

Inicio Reservas Habitaciones Servicios Nosotros Cristina

## Informe

|  |                      |
|--|----------------------|
| Persona a cargo  | Número de habitación |
| c.garcia.lopez   | 127                  |
| Mascota  | Turno                |
| Rex  | Martes               |
| Fecha  |                      |
| 04/04/2023   |                      |
| Descripción  |                      |
| Ha comido correctamente, defecaciones correctas, paseo correcto.<br>Le he dejado un Kong con chuches |                      |

[Ir a Inicio](#) [Modificar](#)



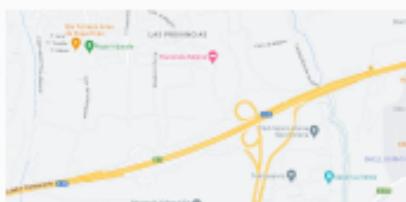


PETLAND RESORT

— ⟲ ⟳ ⟴ ⟵ —

"Una experiencia de lujo para seres de lujo"

## Encuéntranos



Camino Mejina, S/N, 41807 Espartinas, Sevilla

+34 955 123 456

info@petlandresort.com

## Síguenos

— ⟲ ⟳ ⟴ ⟵ —

[f](#) [o](#)

Todos los derechos reservados para Petland Resort 2023

Página para ver y/o editar informe



# Listado de informes

Ordenar por habitación    Ordenar por fecha    Ordenar por mascota    Ordenar por cuidador

Seleccionar    Seleccionar    Seleccionar    Seleccionar

| Habitación: 127 | Mascota: Rex      | Cuidador/a: Cristina | Ver informe |
|-----------------|-------------------|----------------------|-------------|
|                 | Fecha: 04/04/2023 | Turno: Mañana        |             |
| Habitación: 127 | Mascota: Nala     | Cuidador/a: Cristina | Ver informe |
|                 | Fecha: 04/04/2023 | Turno: Mañana        |             |
| Habitación: 127 | Mascota: Salem    | Cuidador/a: Cristina | Ver informe |
|                 | Fecha: 04/04/2023 | Turno: Mañana        |             |

# Encuéntranos

Camino Mejina, 5/N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

Síguenos

f o

Todos los derechos reservados para Petland Resort 2023

Página para listar los informes



The screenshot shows a list of two team members:

- Cristina García López**  
Identificador de empleado: c.garcia.lopez  
Cargo: cuidadora  
Teléfono: 654 321 789  
Email: c.garcia.lopez@petland.com
- Victoria González León**  
Identificador de empleado: v.gonzalez.leon  
Cargo: gerente  
Teléfono: 654 321 987  
Email: v.gonzalez.leon@petland.com

Each entry includes a small profile picture and a "Ver perfil" button.

The section features a map showing the location of Petland Resort at Camino Mejina, S/N, 41807 Espartinas, Sevilla. It also provides contact information:

- Camino Mejina, S/N, 41807 Espartinas, Sevilla
- +34 955 123 456
- info@petlandresort.com

Below the map, there's a "Síguenos" (Follow us) section with social media icons for Facebook and Instagram.

Todos los derechos reservados para Petland Resort 2023

Página para listar empleados



# Listado de clientes

Buscar cliente  Identificador

Identificador del usuario: apergil   
DNI/NIE/Pasaporte: 77564987D   
Teléfono: 654 654 789   
Email: apergil@gmail.com

Ver perfil

Identificador del usuario: mikisanya   
DNI/NIE/Pasaporte: 77564954T   
Teléfono: 600 654 789   
Email: mikisanya@hotmail.com

Ver perfil

# Encuéntranos

PETLAND RESORT

"Una experiencia de lujo para seres de lujo"

Camino Mejina, S/N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

Síguenos

Todos los derechos reservados para Petland Resort 2023

Página para listar clientes



## Listado de usuarios

| Buscar usuario  | Ordenar por rol | Ordenar alfabéticamente    |
|---|-----------------|----------------------------|
| Identificador   | Seleccionar     | Seleccionar                |
| <b>Ana Pérez Gil</b><br>Identificador del usuario: apergil                  | Rol: Cliente    | <a href="#">Ver perfil</a> |
| <b>Miguel Sánchez Yang</b><br>Identificador del usuario: mikisanya          | Rol: Cliente    | <a href="#">Ver perfil</a> |
| <b>Cristina García López</b><br>Identificador del usuario: c.garcia.lopez   | Rol: Empleado   | <a href="#">Ver perfil</a> |
| <b>Victoria González León</b><br>Identificador del usuario: v.gonzalez.leon | Rol: Empleado   | <a href="#">Ver perfil</a> |

## Encuéntranos

Camino Mejina, S/N, 41807 Espartinas, Sevilla  
+34 955 123 456  
info@petlandresort.com

Síguenos

f o

Todos los derechos reservados para Petland Resort 2023

Página para listar todos los usuarios del sistema



## 6. Definición API REST: publicación de servicios

Esta aplicación cuenta cinco rutas distintas desde las que se llevan a cabo acciones. A continuación, se detallan cada uno de los métodos disponibles y su uso.

### 6.1. Index

| INDEX         |        |  |   |
|---------------|--------|--|---|
| Ruta          | Método | Descripción  | Comentario  |
| /             | GET    | Redirecciona a la página index de las vistas   |   |
| /home         | GET    | Redirecciona a la página home de las vistas, que variará en función del usuario autenticado  | Si no hay usuario logado, renderiza a la vista de index |
| /register     | GET    | Redirecciona a la página register de las vistas  |   |
| /reservations | GET    | Redirecciona a la página reservations de las vistas, enviando el listado de mascotas asociadas al usuario logado y la reservas asociadas | Necesita autenticación del usuario                      |
| /rooms        | GET    | Redirecciona a la página rooms de las vistas   |   |
| /services     | GET    | Redirecciona a la página services de las vistas  |   |
| /aboutUs      | GET    | Redirecciona a la página aboutUs de las vistas   |   |
| /statistics   | GET    | Redirecciona a la página statistics de las vistas, enviando un listado de todos las mascotas en el sistema y las reservas                | Acción intermedia, necesita autenticación del usuario   |



## 6.2. Users

| USERS                |        |  |   |
|----------------------|--------|--|---|
| Ruta                 | Método | Descripción  | Comentario  |
| /users               | GET    | Devuelve un listado con todos los usuarios que hay registrados en el sistema ordenados por fecha de registro   | Solo usado por los roles de usuario 'admin' y 'staff'   |
| /users/register      | GET    | Redirecciona a la página que contiene el formulario para registrarse   | Acción intermedia   |
| /users/logout        | GET    | Cierra la sesión del usuario logueado y redirecciona a la página principal   |   |
| /users/editUser      | GET    | Redirecciona a la página que contiene el formulario con los datos actuales del usuario editables (sólo algunos campos)   | Acción intermedia   |
| /users/resetPassword | GET    | Redirecciona a la página que contiene el formulario para cambiar la contraseña   | Acción intermedia   |
| /users/removeUser    | GET    | Redirecciona a la página que contiene el formulario de confirmación para dar de baja al usuario  | Acción intermedia, envía el listado de mascotas y reservas relacionadas con ese usuario   |
| /users/register      | POST   | Inserta un usuario con los datos que se introducen en el formulario. Redirecciona a página de registro en caso de fallo con un mensaje en rojo, en caso de éxito, o a la página de inicio con un mensaje en verde. | Si no hay usuario logado, se hace un registro con valores por defecto para cliente. Si hay usuario logado, se comprueba si su rol es 'admin' o 'staff' y permite escoger el rol que |



|                      |      |  |   |
|----------------------|------|--|---|
|                      |      |  | tendrá dicho usuario  |
| /users/login         | POST | Realiza el registro del inicio de sesión, si la autenticación ha sido correcta, envía al home del usuario, si no, vuelve a la página principal con un mensaje de error | Utiliza la librería de passport   |
| /users/findUser      | POST | Realiza una búsqueda de un usuario usando un formulario donde se introduce el username   | Requiere estar autenticado y ser usuario staff  |
| /users/edit          | PUT  | Modifica los datos del usuario que está logueado (solo se puede hacer sobre el propio usuario)   | Recupera el username de la sesión   |
| /users/editUser      | PUT  | Modifica los datos de un usuario especificado en el formulario   | Solo se puede editar el usuario logado (el propio perfil)                                       |
| /users/resetPassword | PUT  | Modifica la contraseña del usuario, evaluando si está autenticado o no, pues se piden campos adicionales   | La contraseña se almacena hasheada  |
| /users/removeUser    | PUT  | Modifica el status del usuario autenticado para simular su eliminación del sistema. Todas las mascotas y las reservas asociadas también cambian de estado              | La contraseña se cambia, no se elimina el usuario de la base de datos, ya no puede tener acceso |

### 6.3. Pets

| PETS  |        |   |                               |
|-------|--------|---|-------------------------------|
| Ruta  | Método | Descripción                               | Comentario                    |
| /pets | GET    | Devuelve un listado de todas las mascotas | En la vista se controla si el |



|                     |      |   |  |
|---------------------|------|---|--|
|                     |      |   | usuario es un cliente, solo se muestran sus mascotas, y si su rol es de admin o staff devuelve todas las mascotas del sistema  |
| /pets/addPet        | GET  | Renderiza al formulario con los campos para añadir mascota  | Acción intermedia, requiere autenticación del usuario  |
| /pets/editPet       | GET  | Renderiza al formulario con los campos editables de la mascota. Se visualiza con los datos actuales, en editable aquellos que lo son              | Acción intermedia, requiere autenticación del usuario, lleva como parámetro el id de la mascota a editar   |
| /pets/removePet/:id | GET  | Renderiza al formulario donde el usuario simula la eliminación de esta mascota de su perfil   | Acción intermedia, requiere autenticación del usuario, lleva como parámetro el id de la mascota a editar   |
| /pets/addPet        | POST | Añade una mascota con los datos introducidos en el formulario. Se realizan ciertas validaciones tanto en el lado cliente como en el lado servidor | Se requiere que el usuario esté logueado. En caso de ser cliente, se le asigna la mascota a ese usuario, en caso de que el rol de usuario sea admin o staff, se requiere indicar el username del tutor de la mascota |



|                     |      |  |   |
|---------------------|------|--|---|
| /pets/findPet       | POST | Realiza una búsqueda de una mascota usando un formulario donde se introduce el id de mongo asociado a la mascota en cuestión | Requiere estar autenticado y ser usuario staff  |
| /pets/editPet/:id   | PUT  | Modifica los datos de la mascota que se ha indicado en el formulario, solo de los campos editables                           | Requiere que el usuario esté autenticado  |
| /pets/removePet/:id | PUT  | Se realiza el cambio de status a inactive cuando usuario hace el submit, inhabilitando su acceso a esta mascota              | Requiere autenticación del usuario, lleva como parámetro el id de la mascota a editar. No se hace eliminación en la base de datos |

## 6.4. Reservations

| RESERVATIONS                   |        |  |  |
|--------------------------------|--------|--|--|
| Ruta                           | Método | Descripción  | Comentario   |
| /reservations/listReservations | GET    | Devuelve un listado de todas las reservas  | Requiere estar logueado. Si el rol del usuario logueado es admin o staff son todas las del sistema, si no son las de ese cliente |
| /reservations/addReservation   | GET    | Renderiza a la vista con el formulario para crear una reserva, a la que se envía el listado de mascotas relacionadas con el usuario logado | Acción intermedia, requiere que el usuario esté autenticado. Se plantea para que la haga el cliente desde su perfil              |



|                                     |      |  |  |
|-------------------------------------|------|--|--|
| /reservations/generatePDF/:id       | GET  | Permite construir un documento PDF con la información que contiene la reserva que se envía como parámetro  | Requiere que el usuario esté autenticado, utiliza la librería de pdfkit. Se visualiza en una pestaña que se abre en el navegador                           |
| /reservations/editReservation/:id   | GET  | Renderiza a la vista con el formulario con los datos de la reserva que corresponde con la que tenga el id que se envía como parámetro en la url                                | Acción intermedia, requiere la autenticación del usuario   |
| /reservations/removeReservation/:id | GET  | Renderiza a la vista con los datos de la reserva que corresponde con la que tenga el id que se envía como parámetro en la url  | Acción intermedia, requiere la autenticación del usuario   |
| /reservations/addReservation        | POST | Crea una nueva reserva, asociada a un usuario para unas mascotas especificadas   | Requiere estar logueado. Si el rol del usuario logueado es admin o staff se debe especificar el titular de la reserva, si es cliente, este será el titular |
| /reservations/findReservation       | POST | Realiza una búsqueda de una reserva usando un formulario donde se introduce el id de mongo asociado a la reserva en cuestión   | Requiere estar autenticado y ser usuario staff   |
| /reservations/editReservation/:id   | PUT  | Modifica una reserva que corresponda con el id que se recibe como parámetro en la url y siempre que esté activa. Es posible modificar todo menos el identificador y el titular | Requiere estar logueado  |



|                                     |        |   |  |
|-------------------------------------|--------|---|--|
| /reservations/removeReservation/:id | DELETE | Elimina una reserva siempre que esté activa | Requiere estar logueado y que la fecha de la reserva no sea anterior a la actual. Se elimina de la base de datos |
|-------------------------------------|--------|---|--|

## 6.5. Datas

| DATAS               |        |   |  |
|---------------------|--------|---|--|
| Ruta                | Método | Descripción   | Comentario   |
| /datas/economy      | GET    | Devuelve un array de documentos con el sumatorio de los importes de las reservas y la fecha de checkin                        | Requiere estar logueado. Solo es visible para admin y manager.<br>Estos datos serán pintados en gráficas |
| /datas/roomType     | GET    | Devuelve un array de documentos con el sumatorio de los tipos de habitaciones y la cantidad de reservas que hay para cada uno | Requiere estar logueado. Solo es visible para admin y manager.<br>Estos datos serán pintados en gráficas |
| /datas/reservations | GET    | Devuelve un array de documentos con el sumatorio de reservas de cada mes  | Requiere estar logueado. Solo es visible para admin y manager.<br>Estos datos serán pintados en gráficas |
| /datas/petSpecie    | GET    | Devuelve un array de documentos con el sumatorio de los tipos de mascotas registradas y la cantidad que hay de cada uno       | Requiere estar logueado. Solo es visible para admin y manager.<br>Estos datos serán pintados en gráficas |



## 7. Manual de despliegue

Para que la aplicación pueda ser utilizada de forma real es necesario desplegarla de manera que sea accesible y utilizable. En el caso de esta aplicación, utilizaremos Railway como servidor para el despliegue automático. A continuación, se detallan los pasos a seguir. Es fundamental tener en cuenta que para hacer este tipo de despliegue y uso de la aplicación web se necesita acceso a internet, un navegador y, en este caso, tener el código de la aplicación en un repositorio de GitHub.

### 7.1. Archivos de configuración

- **Actualización de las dependencias:** ejecutamos el comando `npm install` desde el directorio raíz de nuestro proyecto para instalar y actualizar todas las dependencias de nuestro proyecto.

```
● PS C:\Users\Andrea\Documents\GitHub\PetlandResort-ProyectoFinal\src> npm install
  up to date, audited 251 packages in 734ms
  48 packages are looking for funding
    run `npm fund` for details
  found 0 vulnerabilities
○ PS C:\Users\Andrea\Documents\GitHub\PetlandResort-ProyectoFinal\src>
```

- **Archivo "start.sh":** creamos este archivo en el directorio raíz del proyecto. En este archivo incluiremos el mando que se debe ejecutar al iniciar la aplicación.

```
#!/bin/bash
npm start
```

- **Archivo "railway.yml":** también, en el directorio raíz del proyecto, creamos este archivo, que se encargará de ejecutar el comando que indicamos en el archivo anterior y hacer las configuraciones necesarias.

```
version: '1'
services:
  app:
    build:
      env:
```

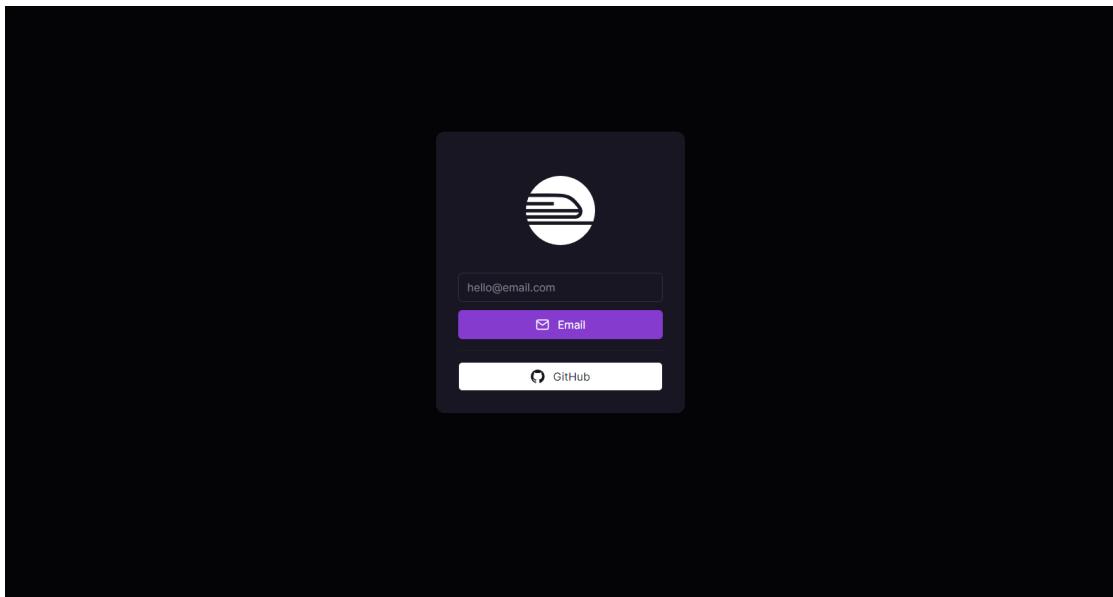


```
PORT: 3000  
command: ./start.sh
```

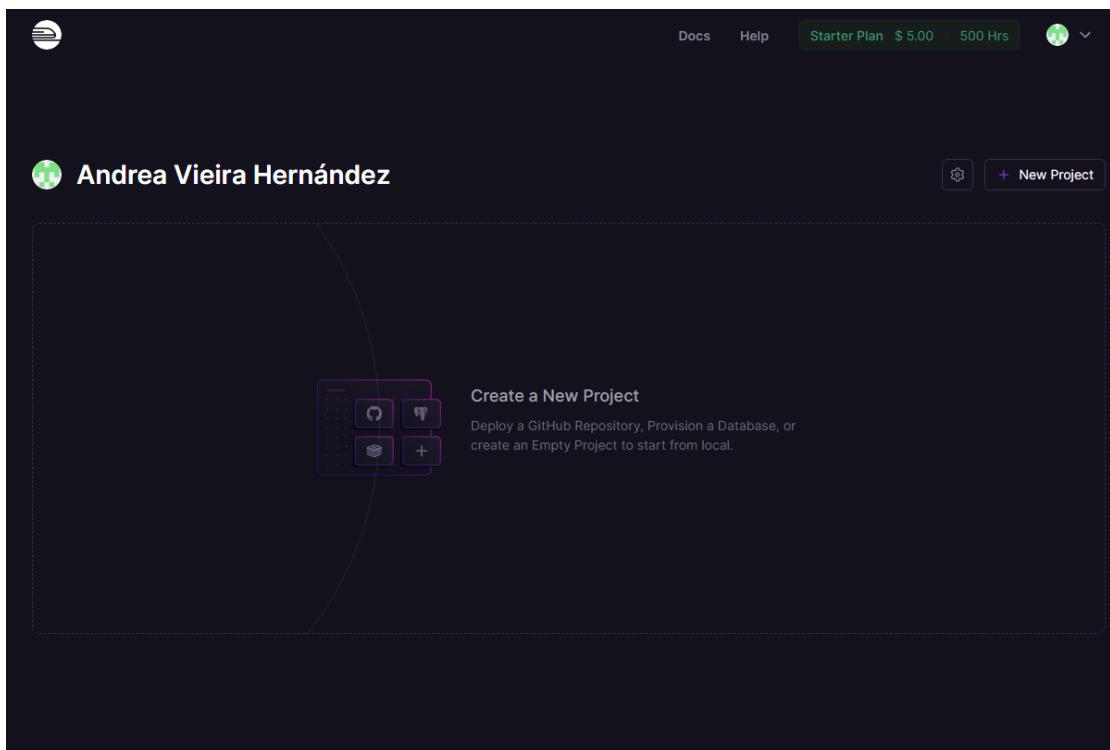
## 7.2. Configuración de la plataforma de despliegue Railway

Tras haber configurado nuestro proyecto, con los archivos necesarios para la automatización del despliegue, procedemos a enlazar nuestra aplicación con los servicios de Railway.

- **Registro en la plataforma de Railway a través de su web y login:**  
En este caso, resulta muy cómodo hacer el login a través de la cuenta de GitHub.  
Accedemos al enlace: <https://railway.app/dashboard> e indicamos la opción más conveniente, en este caso GitHub.

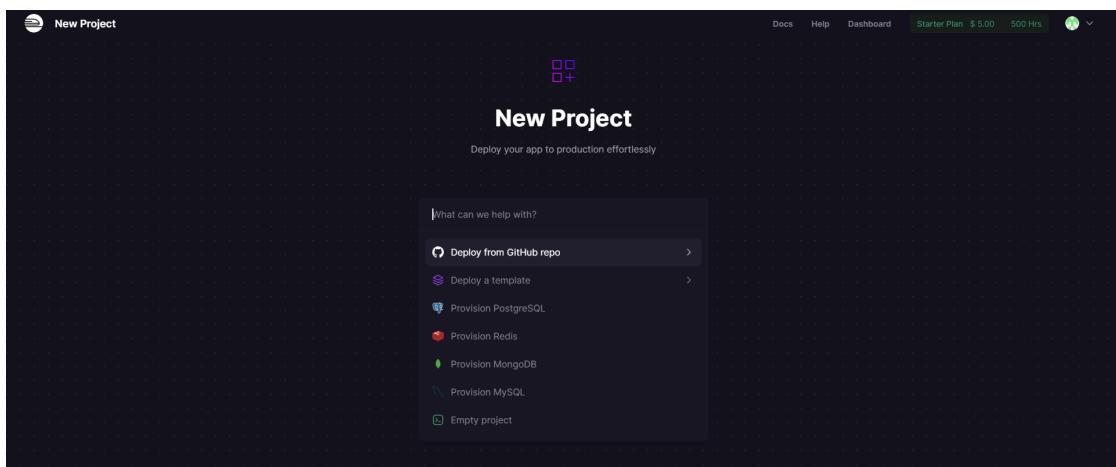


- **Creación de un nuevo proyecto en el dashboard:**  
En nuestro dashboard encontramos el botón "+New Project" y se nos abre la ventana de configuración del proyecto.



- **Selección del tipo de despliegue:**

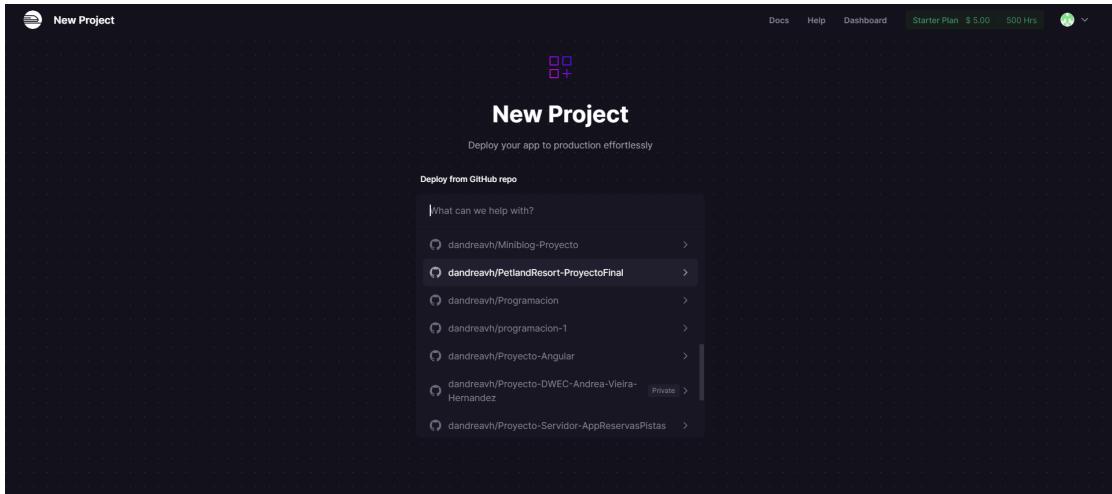
Lo primero que debemos indicar es el tipo de despliegue. En nuestro caso, queremos que se haga de forma automática a través de un repositorio de nuestro GitHub, de manera que cada vez que se haga “push” en un commit lanza el despliegue nuevamente.





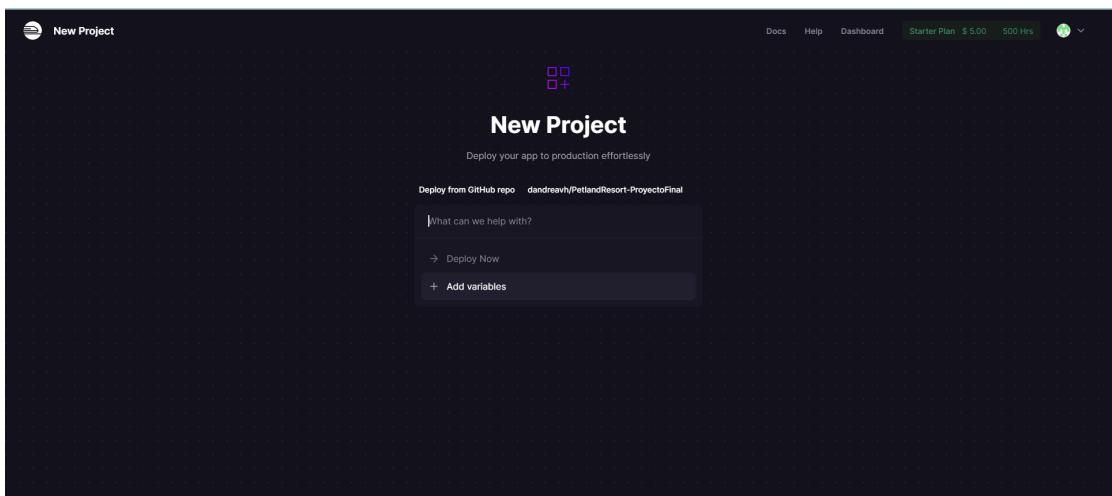
- **Selección del repositorio del proyecto:**

Cuando ya el sistema sabe que va a utilizar un repositorio de GitHub, el siguiente paso es referenciar el repositorio del proyecto que queremos desplegar.



- **Opción para introducir las variables:**

En este paso, Railway nos da la opción de desplegar directamente o de añadir las variables del proyecto, en este caso, como nuestro proyecto cuenta con variables almacenadas en el archivo .env, es necesario añadirlas, pues está una de las más importantes, la url que conecta con nuestra base de datos. Es óptimo tener las variables en archivos que preserven la seguridad, pues contienen datos sensibles.





- **Añadir variables:**

Se nos abre la ventana de configuración de variables, donde vamos introduciendo una a una las que necesitamos. En este caso, tenemos el puerto (aunque no es necesario), el entorno y, la más importante, la uri de la base de datos. Se aprecia cómo Railway oculta también estos datos, pues son sensibles.

The screenshot shows the Railway interface for the 'PetlandResort-ProyectoFinal' service. On the left, there's a deployment status card for 'PetlandResort-Proyecto...' showing a failure. In the center, the 'Variables' tab is selected, displaying three environment variables: 'DB\_URI', 'NODE\_ENV', and 'PORT'. On the right, a modal window titled 'Redeploying Service' is open, stating 'Redeploying PetlandResort-ProyectoFinal now'.

- **Configuraciones del entorno:**

En este paso, confirmamos la rama desde la que queremos hacer el despliegue y procedemos a obtener la url que nos permitirá hacer uso de nuestra aplicación a través de la conexión de internet. En nuestro caso, al no tener un dominio propio, necesitamos que nos cree uno.

Con este enlace, podemos acceder a nuestra aplicación:  
<https://petlandresort-proyectofinal-production.up.railway.app/>

The screenshot shows the Railway interface for the 'PetlandResort-ProyectoFinal' service under the 'Settings' tab. The 'Service' section indicates settings are scoped to 'ALL environments'. The 'Domains' section allows exposing the service to the public internet via a generated or custom domain. A 'Check Suites' section is also present.



### - **Configuraciones general del servicio:**

Si nos desplazamos hacia abajo, encontramos la configuración del servicio. En las configuraciones generales, tenemos que comprobar que el repositorio sea el adecuado y, ahora, lo más importante, indicar el directorio raíz de nuestro proyecto. En este caso, como en el directorio raíz tenemos también la carpeta para la documentación, debemos indicar que nuestra aplicación se encuentra en la carpeta /src, siendo esta la raíz.

The screenshot shows the Vagabond-Spy deployment interface. At the top, there's a navigation bar with 'Docs', 'Help', 'Starter Plan \$ 5.00', '500 Hrs', and a user icon. Below the navigation, the service name 'PetlandResort-ProyectoFinal' is displayed. The 'Settings' tab is selected. Under 'Service Name', it shows 'PetlandResort-ProyectoFinal'. Under 'Source Repo', it shows 'dandreasvh/PetlandResort-ProyectoFinal'. Under 'Root Directory', it says 'Directory that will be used for build and deployment steps. Additionally, builds will be skipped if no files changed within this directory.' and has a field containing '/src'. At the bottom, there are tabs for 'Build' and 'Builder'.

### - **Configuraciones del despliegue:**

Seguimos en la misma pestaña de configuración y más abajo encontramos el apartado referente al despliegue, en este punto, debemos indicar el comando para que arranque nuestra aplicación, que será, en principio, el mismo que utilizamos para levantar el servicio desde nuestro localhost, es decir, `npm run start:dev`.



The screenshot shows the Vagabond-Spy dashboard for the service 'PetlandResort-ProyectoFinal'. On the left, there's a small preview window showing a GitHub commit message: 'PetlandResort-Proyecto... petlandresort-proyectofinal-... 2 minutes ago via GitHub'. The main panel has tabs for Deployments, Variables, Metrics, and Settings. Under Settings, there's a 'Watch Paths' section with a text input containing '/src/src/\*\*'. Below that is a 'Deploy' section with a 'Start Command' input set to 'npm run start:dev'. In the bottom right corner of the main panel, there's a 'Redeploy Scheduled' button with a tooltip: 'We will redeploy PetlandResort-ProyectoFinal when you're done'.

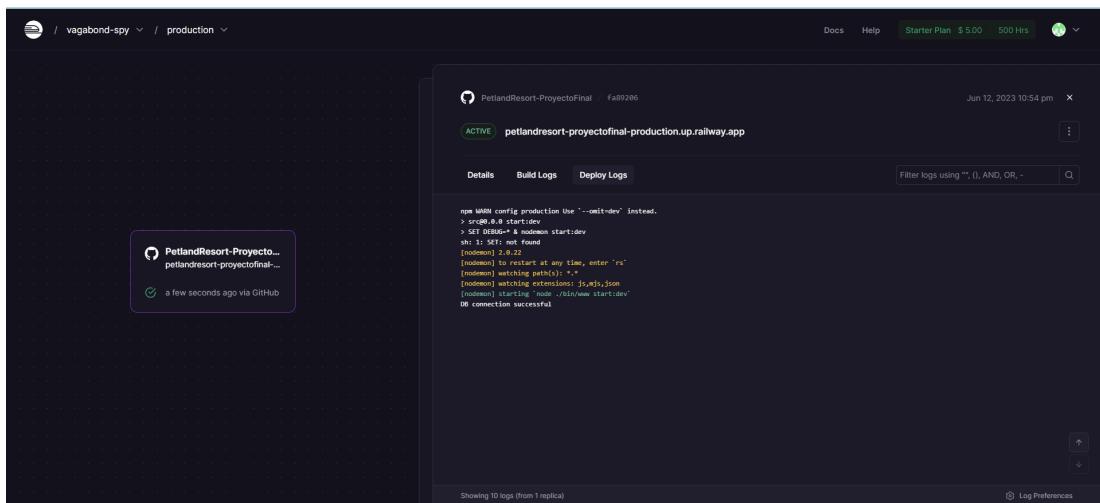
#### - **Historial de despliegue:**

En nuestro dashboard tenemos, por un lado, la ventana con el estado actual de nuestro servicio (lazo izquierdo), y del lado derecho en grande la ventana con todos los detalles que hemos visto en los pasos anteriores. Aquí es importante tener un control del historial de despliegue, pues de esta manera podemos encontrar “roturas” en nuestra aplicación.

The screenshot shows the Vagabond-Spy dashboard focusing on the deployment history for 'PetlandResort-ProyectoFinal'. It lists three recent events: 1. A successful deployment 'petlandresort-proyectofinal-production.up.railway' made a minute ago via GitHub, with a status of 'AboutUs bug' and a 'main' branch. 2. A failed deployment 'Removed' made 3 minutes ago via GitHub, also with a 'main' branch. 3. Another failed deployment 'Failed' made 5 minutes ago via GitHub, with a 'main' branch. Each event has a 'View Logs' button to its right.

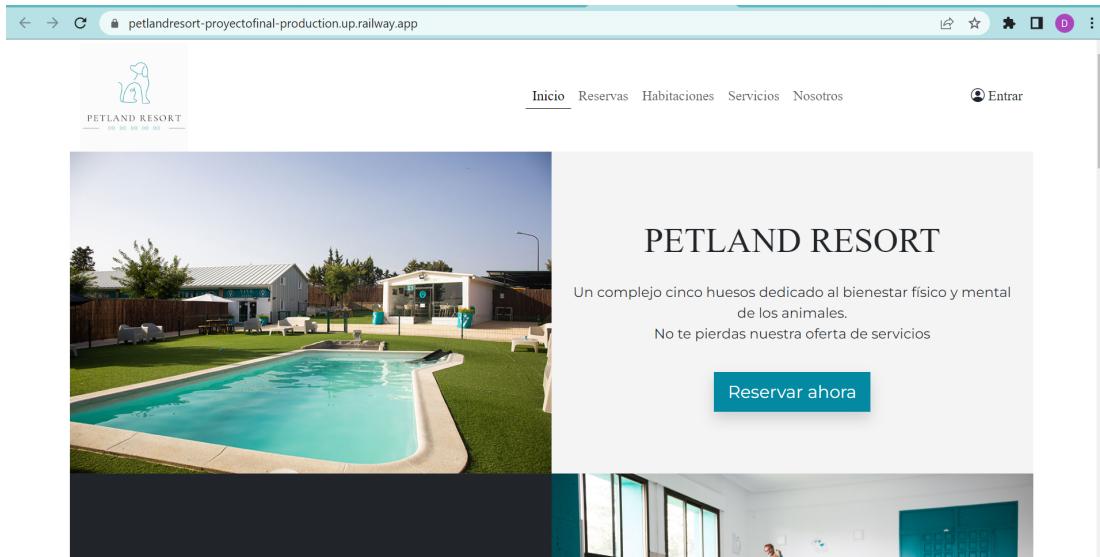
#### - **Logs del despliegue:**

Para cada uno de los lanzamientos de despliegue automático que se hace al hacer commit, tenemos a nuestra disposición los logs del servidor, lo cual nos permite tener acceso al flujo de acciones de nuestra aplicación y, en caso de errores, poder entender en qué punto se producen. Los datos que nos proporciona son los que veríamos en nuestra terminal local.



### - Comprobación del acceso a la aplicación:

Tras observar que el estado del despliegue sea “active”, comprobamos que tenemos acceso a través del enlace que nos ha generado automáticamente el servicio.





## 8. Postmortem y conclusiones

La aplicación web de Petland Resort ha visto la luz en su primera versión. Tras tres meses de arduo trabajo, se ha logrado una aplicación funcional y completa, que cumple con los objetivos planteados y que ha pretendido integrar los conocimientos adquiridos a lo largo del ciclo formativo.

El proyecto ha sido motivado tanto por aspectos tecnológicos como por su temática. Se ha utilizado un stack completo JavaScript en formato monolítico, aprovechando su popularidad y flexibilidad. Se ha cuidado la estética y la usabilidad de la aplicación, teniendo en cuenta la experiencia de usuario.

Llegar a este resultado no ha sido un camino fácil. Aunque gracias a las tecnologías modernas empleadas se ha contado con potentes librerías y opciones para facilitar el desarrollo, el hecho de que una sola persona realice el ciclo de vida completo de una aplicación es una tarea compleja, sobre todo, cuando se tiene que someter a juicio y pruebas un producto hecho por nuestras propias manos.

No obstante, ha sido una experiencia muy enriquecedora tanto para las destrezas tecnológicas, como para las estimaciones, requerimientos y elementos a tener en cuenta. Durante su desarrollo, se han tenido que superar diversos desafíos y se han tomado decisiones importantes en cuanto a estrategias y prioridades o rediseños del planteamiento inicial.

Como hemos comentado anteriormente, esta “release” se trata de una primera versión de la aplicación, pues requiere de mejoras para los elementos ya existentes y de adición de funcionalidades que respondan a la realidad de este negocio ficticio.

Por un lado, será necesario añadir una lógica para la gestión de habitaciones, los servicios detallados y los informes generados en cada turno. Asimismo, la empresa necesitaría una cuenta de google para empresas, pues de esta manera se puede conectar el servicio de envío de correos, que ya se deja planteado en el código. Además, será importante manejar filtros de búsqueda para que estas sean más rápidas y cómodas, sobre todo de cara a tener un gran volumen de datos.

Sería interesante, también, contar con un sistema de autenticación a través de las plataformas habituales como google, pudiendo enlazar el calendario donde añadir la notificación de inicio de reserva.



De cara a la gestión interna del centro, como mejoras y ampliaciones se podría destacar el hecho de contar con un sistema para ver la disponibilidad de habitaciones para el día, mes y semana.

Teniendo en cuenta el tiempo a dedicar, la curva de aprendizaje y la experiencia en hacer un proyecto de esta índole, el foco de acción se ha apuntado a las funcionalidades principales para que el producto pueda salir al mercado. Si hacemos una estimación muy generalizada, suponiendo que se han invertido 200 horas para la realización del proyecto y con un valor de 30€ la hora de trabajo de un desarrollador, el costo de este proyecto, a este nivel sería de unos 6.000€.

En resumen, a pesar de ser una primera versión, la puesta en marcha de la aplicación web de Petland Resort ha sido exitosa en términos de funcionalidad y cumplimiento de objetivos. Las mejoras propuestas permitirán que la aplicación se adapte aún mejor a las necesidades del negocio y brinden una experiencia de usuario más completa. Con un enfoque en las funcionalidades principales, el proyecto ha sentado las bases para un producto sólido que puede ser lanzado al mercado y con perspectiva de seguir evolucionando en el futuro.