



## MÓDULO: PROYECTO

	<b>ALUMNO/A:</b> Andrés Felipe Rivera Riaño
	<b>CURSO:</b> 2021/2022
	<b>CICLO:</b> Administración de Sistemas Informáticos en Red
	<b>FAMILIA PROFESIONAL:</b> Informática y Comunicaciones RD 1629/2009
<b>RESPONSABLE/S DEL SEGUIMIENTO</b>	José Ignacio Prieto Tinoco
<b>TITULO DEL PROYECTO</b>	Desarrollo de una API en Laravel bajo un entorno virtualizado
<b>DESCRIPCIÓN DEL PROYECTO</b>	Creación, instalación, puesta en marcha de un servidor virtualizado CentOS7, que sirve a una aplicación desarrollada con el “framework” Laravel. Esta aplicación, en la parte de “back-end”, constará de una API que devolverá la información de una base de datos previamente creada y que, mediante las necesidades del desarrollador de “front-end”, cumplirá las funciones siguientes: mostrar, crear, actualizar y eliminar registros de las bases de datos. También constará de una interfaz de autenticación donde podrán registrarse e iniciar sesión los distintos usuarios que accedan a la aplicación web.

## INDICE

<b>1. INTRODUCCIÓN</b>	4
<b>2. ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO</b>	7
<b>3. OBJETIVO GENERAL DEL PROYECTO.</b>	9
<b>4. OBJETIVOS ESPECÍFICOS DEL PROYECTO.</b>	11
<b>5. TAREAS.</b>	14
Tarea 1 - VMWare ESXi e implementación del servidor CentOS 7.	15
Tarea 1.1 - Acceso al servidor.	15
Tarea 1.2 - Creación de la máquina virtual.	17
Tarea 1.3 - Iniciar y configurar la máquina virtual.	21
Tarea 1.4 - Configuración de la pila LAMP.	30
Tarea 1.4.1 -Instalación de Apache 2.4.6.	32
Tarea 1.4.2 -Instalación de MySQL (MariaDB) 10.4.24.	35
Tarea 1.4.3 -Instalación de PHP 8.1.4.	37
Tarea 1.4.4 -Instalación de phpMyAdmin 5.1.3-2.el7.remi (opcional).	40
Tarea 1.4.5 - Configuración de SELINUX y Firewall.	43
Tarea 2 - Instalación de Laravel.	46
Tarea 3 - Instalación y configuración de ProFTPD y Filezilla.	50
Tarea 3.1 - ProFTPD, SELinux, Firewall y Passive FTP.	50
Tarea 3.2 - Filezilla.	54
Tarea 4 - Desarrollo de una API en local.	57
Tarea 4.1 - Preparación del proyecto.	57
Tarea 4.2 - Diseño de la bases de datos.	59
Tarea 4.3 - Gestión de las bases de datos.	60
Tarea 4.4 - Inserción de datos de prueba para la base de datos.	62
Tarea 4.5 - Creación de la API e introducción a POSTMAN.	73
Tarea 5 - Tests realizados con Postman.	85
Tarea 5.1 - Test de “aircraft”.	85
Tarea 5.2 - Test de “specifics”.	87
Tarea 5.3 - Test de “trainers”.	89
Tarea 5.4 - Test de “trainees”.	91

Tarea 5.5 - Test de “chapters”.	93
Tarea 5.6 - Test de “products”.	95
Tarea 5.7 - Test de “projects”.	97
Tarea 5.8 - Test de “logbooks”.	99
Tarea 5.9 - Test de “join_trainers_logbooks”.	101
Tarea 5.10 - Test de “join_trainees_logbooks”.	103
Tarea 5.11 - Test de “topics”.	105
Tarea 5.12 - Test de “sections”.	107
Tarea 5.13 - Test de “tasks”.	109
Tarea 5.14 - Test de “exam_tasks”.	111
Tarea 5.15 - Test de “model_doc_refs”.	113
Tarea 5.16 - Test de “logbook_entries”.	115
Tarea 5.17 - Test de “trainee_notes”.	117
Tarea 5.18 - Test de “trainer_notes”.	119
Tarea 5.19 - Test de “trainee_doc_refs”.	121
Tarea 5.20 - Test de “evaluations”.	123
Tarea 6 - Configuración inicial de autenticación.	125
Tarea 7 - Subida de la aplicación al servidor.	126
<b>6. RECURSOS HUMANOS.</b>	<b>130</b>
<b>7. RECURSOS MATERIALES.</b>	<b>132</b>
<b>8. CRONOGRAMA.</b>	<b>134</b>
<b>8. BIBLIOGRAFÍA.</b>	<b>136</b>

## 1. INTRODUCCIÓN

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

## 1. Introducción.

Como parte del programa Erasmus de centro IES La Marisma, tuve la oportunidad de hacer mi formación en centros de trabajo en una empresa de desarrollo de software en Irlanda, llamada ASTech.

Dentro de las tareas encomendadas fueron las siguientes:

- Creación y configuración de un servidor virtual CentOS 7 en diversos hipervisores, como VirtualBox o VMWare.
- Puesta en marcha y preparación de la máquina virtual para servir aplicaciones web en modo local.
- Creación de una API que sirva para poder crear y desarrollar una aplicación web.

Esta aplicación web lleva el nombre de Elisa, y es en esencia una aplicación alojada en un entorno local, que permite a los entrenadores de las escuelas de aeronáutica llevar el seguimiento de tareas asignadas a los aprendices.

A lo largo del proyecto se harán uso de diversas tecnologías y sistemas operativos donde podemos destacar la pila LAMP en una máquina virtual, el uso de Laravel como framework elegido para poder servir al “back-end” de la aplicación y su implementación junto con las bases de datos en MySQL, que, en conjunto, harán uso de una API que arrojará datos en formato JSON, es decir, una aplicación que permita trabajar con una base de datos y después poder consumirla mediante cualquier otro lenguaje o un framework “front-ed”, como podría ser VUE.js, Angular entre otros.

## 2. ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Curso 2021/2022.

## 2. Origen y contextualización del proyecto.

ASTech es una empresa de software con sede en Irlanda. Se especializa en el desarrollo de software para la capacitación en la industria de la aviación. Su producto insignia, GEMS, es una aplicación web utilizada para la administración de exámenes para pilotos, ingenieros de mantenimiento, tripulantes de cabina y demás personal de las aerolíneas. GEMS se entrega como una pila LAMP y se aloja en la nube o internamente en los sitios de los clientes. Los exámenes se entregan en papel o en computadoras o tabletas, siendo estas, aplicaciones multiplataforma (iPadOS, iOS, Android, Windows, Linux, MacOS).

Como parte de la investigación y el desarrollo en curso, ASTech está trabajando en una serie de otros proyectos. Uno de ellos es la llamada aplicación Elisa, que los ingenieros aeronáuticos utilizarán para realizar un seguimiento de sus tareas de formación. La idea es que un ingeniero tenga todas sus tareas registradas en un iPad y este se sincronice periódicamente con una aplicación de “back-end”.

En este proyecto integrado nos centraremos en la configuración, la API y la autenticación de Laravel. La configuración de la máquina virtual CentOS 7 que se ejecuta en un servidor VMWare ESXi y la configuración inicial de Laravel.

En principio, la aplicación se desarrollará internamente, para que luego se pueda probar en el servidor virtual, donde posteriormente se ejecutarán los tests pertinentes con distintas aplicaciones informáticas y así, poder corregir errores antes de subirla a producción y por tanto, ofrecersala a los clientes como producto final.

### **3. OBJETIVO GENERAL DEL PROYECTO.**

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

### **3. Objetivo general del proyecto.**

La tecnología está cambiando el estilo de la educación a un ritmo cada vez mayor. Atrás han quedado esas pilas de libros, montones de documentos apilados y aquellos métodos tradicionales de enseñanza como lo son las pizarras convencionales. El tiempo va pasando y, con ello, han habido muchísimos avances en el ámbito de la enseñanza y el aprendizaje.

Las escuelas de formación de aviadores van buscando esa modernización, y junto a las tecnologías más punteras, quieren dejar atrás todas esas convenciones. Necesitan la sustitución de los libros de tareas a los que se ataban para poder hacer el seguimiento y validación de aquellas tareas asignadas a los alumnos que están formando. El beneficio que esto conlleva, es tanto a nivel económico como a nivel formativo y medioambiental.

Se plantea una solución en base a una aplicación multiplataforma compatible con dispositivos móviles, como tablets o iPads, en las que los estudiantes tendrán su cuaderno de tareas donde se le asignarán cada una de las tareas necesarias para poder completar su formación, y, desde la tablet del formador, poder revisar y hacer constancia de las tareas realizadas.

En la primera fase del proceso se implementará la máquina virtual encargada de servir la API (Application Programming Interface) o Interfaz de Programación de Aplicaciones, que permita el flujo de datos desde el servidor, donde quedan almacenados todos los registros correspondientes, y en las tablets de los estudiantes y formadores, de una aplicación web que se ve servida de esa API.

Esta aplicación web será multiplataforma y permitirá a la tablet almacenar en caché los resultados hasta su sincronización con el servidor, útil en aquellas ocasiones donde no se disponga de conexión a la red y así seguir el concepto de API-First.

## 4. OBJETIVOS ESPECÍFICOS DEL PROYECTO.

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

### **3. Objetivos específicos del proyecto.**

Los objetivos específicos a tratar en el proyecto serán:

- 1. Creación de una máquina virtual en un Hipervisor.**

En este apartado se desarrollarán los conocimientos relacionados con la administración de sistemas operativos y de redes. Habrá que administrar los recursos disponibles del sistema así como la puesta en marcha e instalación de un sistema operativo que permita actuar de servidor de nuestra aplicación. Se basará en CentOS 7 y albergará la aplicación de back-end. Se dispondrá a configurar los permisos, las conexiones e interfaces de red así como los distintos servicios de manera segura para que sólo las personas autorizadas puedan administrarla.

- 2. Configuración del servidor LAMP.**

LAMP responde a Linux, Apache, MySQL (Maria-DB) y PHP. Habrá que instalar, configurar y gestionar cada uno de los elementos mencionados para que nuestra API sea funcional y segura. Es una parte básica y esencial del proyecto y hay que asegurarse que funciona correctamente.

- 3. Configuración e instalación de Laravel y de la base de datos.**

Laravel será el framework de PHP donde se desarrollará nuestra API. Se aprenderá los principios básicos de Modelo-Vista-Controlador que permitirán a la aplicación front-end consumir de esa API. Se procederá a la instalación de los componentes necesarios para que Laravel pueda ser ejecutado desde el servidor.

- 4. Creación de la base de datos y la API.**

Mediante las herramientas que nos proporciona Laravel, crearemos la base de datos correspondiente a todos los campos que se requieran para la aplicación Elisa. Habrá que asegurarse que todas las tablas estén correctamente diseñadas para que, a la hora de la migración, estas se efectúen correctamente.

A partir de ellas, crearemos la API en base al diseño anterior, que permitirá hacer CRUD (Create, Read, Update y Delete), es decir, creación, lectura, actualización y borrado de registros, algo necesario para poder interactuar con la aplicación front-end que esté consumiendo de la API. Los datos se consumen en formato JSON, que es un Formato ligero de intercambio de datos, optimizado sobre todo para aplicaciones front-end en Javascript como Vue.js. También se creará la interfaz de autenticación de usuarios.

##### 5. Pruebas de la API con Postman.

Se procederá a realizar pruebas de funcionamiento de la API con Postman, una herramienta dirigida a desarrolladores web que permite realizar peticiones HTTP a cualquier API. Se procederá a hacer CRUD de cada una de las tablas y documentarlas mediante capturas de pantalla.

## 5. TAREAS.

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

## 5. Tareas.

En este apartado se define tanto el objetivo que persigue la ejecución de cada una de las tareas necesarias, así como la definición de las mismas.

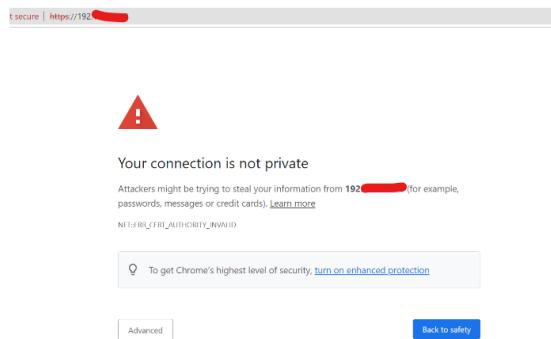
### Tarea 1 - VMWare ESXi e implementación del servidor CentOS 7.

VMware ESXi (Elastic Sky X integrado) es un hipervisor tipo 1 o “bare-metal” de clase empresarial desarrollado por VMware para implementar y brindar servicio a computadoras virtuales. Esto significa que se ejecuta directamente en el hardware físico de la máquina host, es decir, no tiene que cargar un sistema operativo subyacente antes de eso. Debido al hecho de que los hipervisores tipo 1 tienen acceso directo al hardware subyacente (y no a otros sistemas operativos y controladores de dispositivos con los que contentar), este tipo de hipervisor se considera el más eficiente y el que ofrece mejor rendimiento para la informática empresarial.

#### Tarea 1.1 - Acceso al servidor.

Para acceder al servidor alojado en la sede de ASTech, debemos abrir un navegador web dentro de la red local y escribir la dirección local correspondiente. Ejemplo: 192.xxx.xxx.xxx

Recibiremos una advertencia de seguridad. En este caso, al no estar alojado en ningún DNS, el navegador no puede verificar el certificado de seguridad, ya que al ser un recurso local, no lo tiene. Pero no es un problema, ya que sabemos de dónde viene la fuente.



Una vez dentro del navegador, accederemos a las opciones avanzadas.

El error «La conexión no es privada» suele aparecer cuando el navegador detecta un problema al crear una conexión SSL o no puede verificar el certificado. Debido a que nuestra dirección es local, por defecto el navegador lanza esa advertencia. Si la implementación de nuestra red es correcta, no debería haber problemas en acceder a la misma.

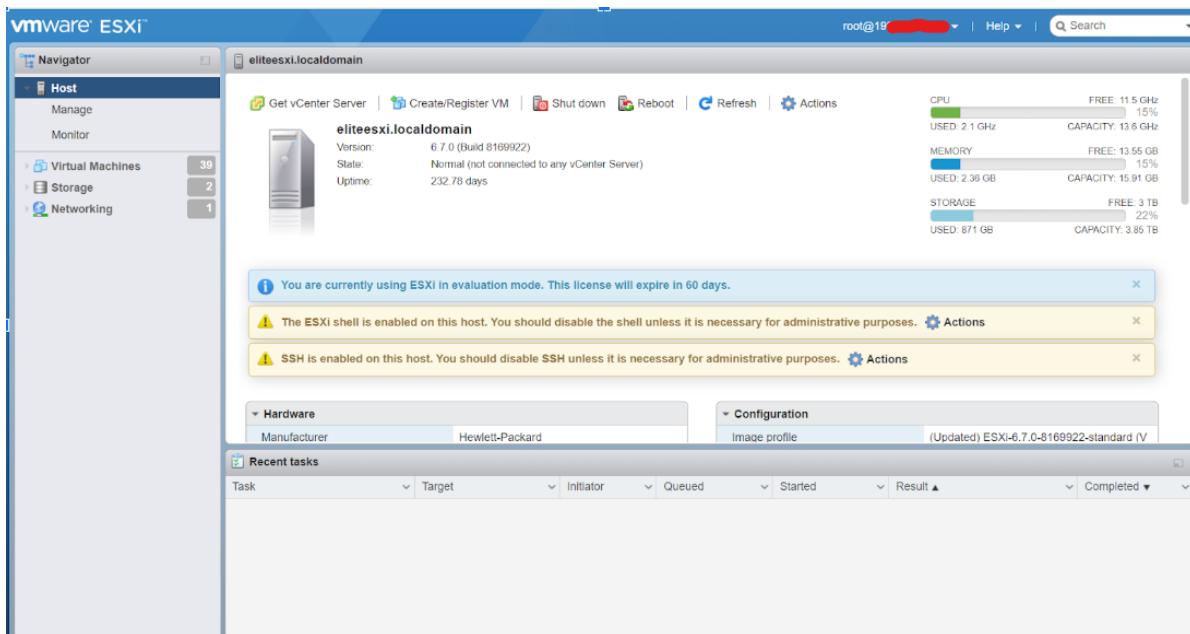
En el caso de que salga, habrá que proceder al acceso mediante las opciones avanzadas > Acceder a la dirección IP local.

Una vez dentro, nos cargará la interfaz de autenticación de nuestro servidor VMWare.

Dentro de la interfaz, tendremos que iniciar sesión con las credenciales proporcionadas por la empresa o el administrador de sistemas que gestione el servidor.



Una vez realizado el proceso ya estaríamos en la interfaz web donde podemos gestionar nuestro servidor y crear nuestra máquina virtual. Esta sería la ventana principal tan pronto como inicie sesión.



## Tarea 1.2 - Creación de la máquina virtual.

Ahora vamos a continuar con el proceso de creación de una máquina virtual en nuestro servidor.

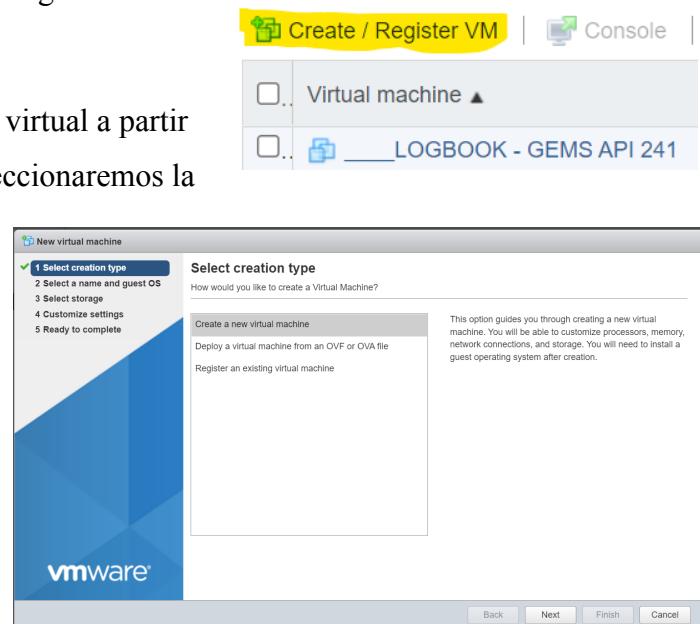
En el lado derecho de nuestra interfaz, hacemos clic en Máquinas virtuales.

Nos mostrará una lista de máquinas virtuales ya implementadas así como los elementos que nos permitirán crear una nueva máquina.

A continuación vamos a la opción “Crear/registrar máquina virtual”:

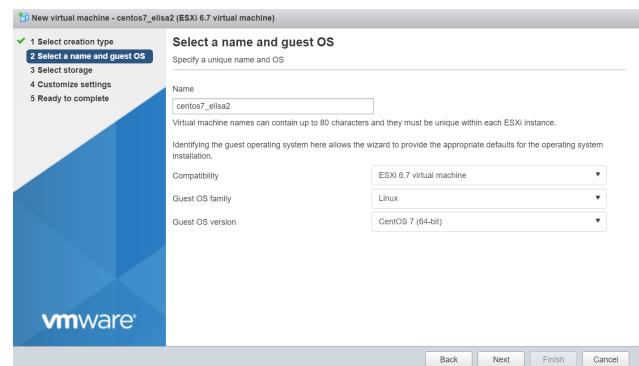
Como vamos a crear una nueva máquina virtual a partir de una ISO previamente descargada, seleccionaremos la opción “Crear una nueva máquina virtual”:

Hacemos clic en Siguiente.



Seleccionamos el nombre, la compatibilidad, el tipo de sistema operativo y la versión del sistema operativo:

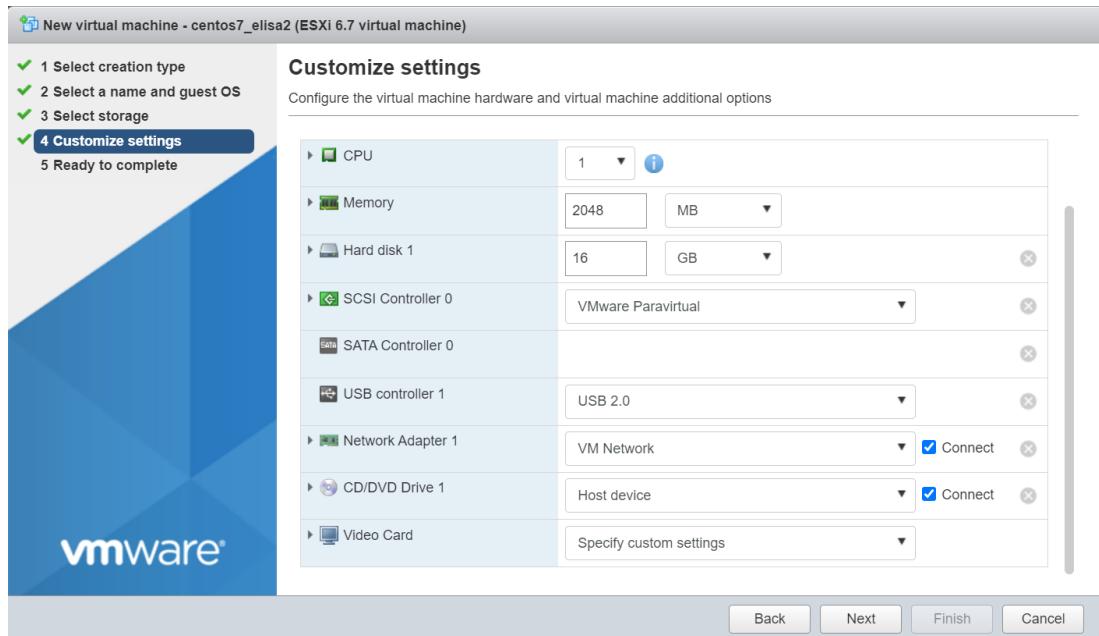
En este caso tenemos que ajustar los parámetros para que sea compatible con CentOS 7.



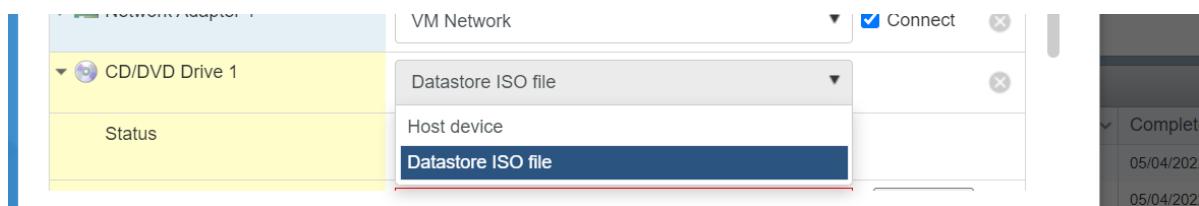
Hacemos clic en Siguiente.

Luego seleccionamos la cantidad de memoria, procesador, disco duro, etc. que usaremos para nuestra máquina virtual.

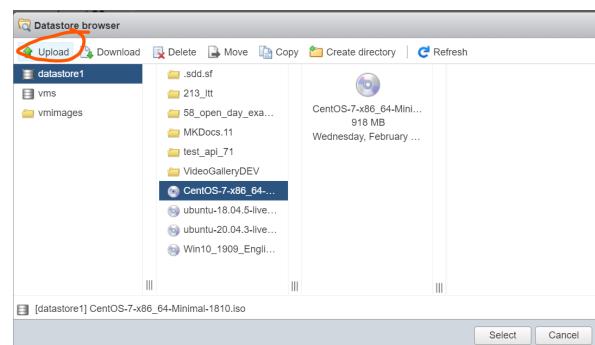
Dejaremos los valores por defecto, ya que, en principio, podemos aumentar las capacidades en el futuro si se requiere.



No obstante, en el apartado “Unidad de CD/DVD” seleccionaremos la opción “Archivo ISO del almacén de datos” del menú desplegable.

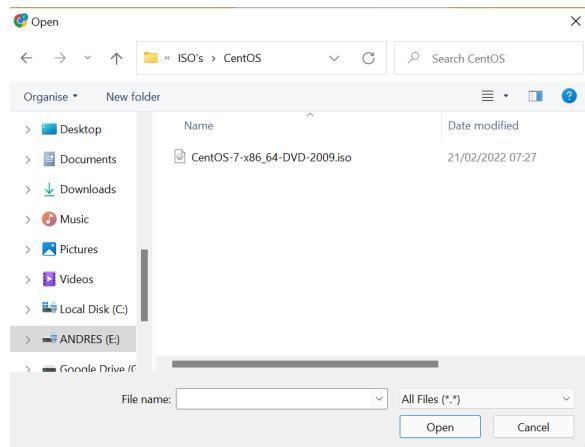
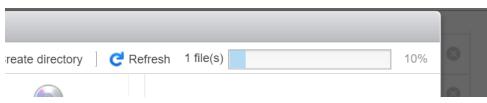


Ahora vamos a cargar nuestra ISO específica, habrá algunas ya cargadas, pero a veces necesitamos una que no lo esté. Damos clic en “Subir”.



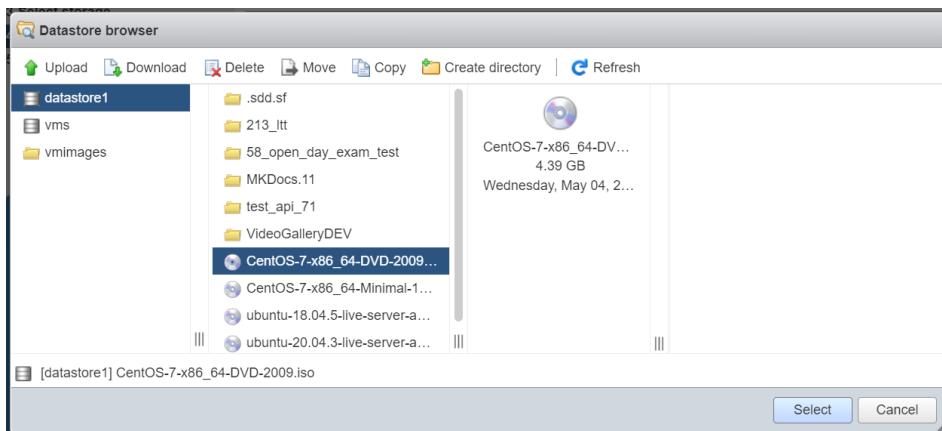
Buscamos el archivo ISO en nuestro ordenador, donde tenemos almacenada la descarga para posteriormente subirla.

Esperamos a que se complete la carga



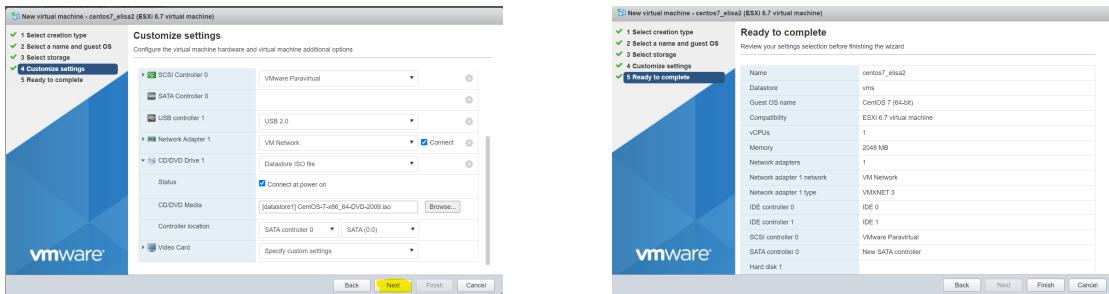
Una vez finalizada la carga veremos como aparece dentro del directorio ISOS que habíamos creado. Ahora sí, ya tenemos el primer sistema operativo dentro de nuestro “Datastore” de VMWare ESXi que usaremos para iniciar nuestra máquina virtual.

Hacemos clic en Seleccionar.

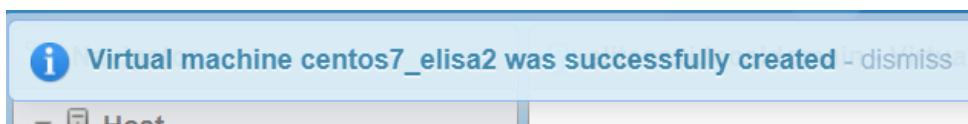


Esto nos permitirá cargar todos los recursos que se instalarán en nuestro disco duro virtual, que hará arrancar el sistema desde nuestra máquina virtual.

Volviendo a la ventana anterior, hacemos clic en "siguiente". Finalmente veremos el resumen de todas las opciones seleccionadas y damos clic en “Finalizar”.



Si todo salió bien, obtendremos una advertencia como esta:



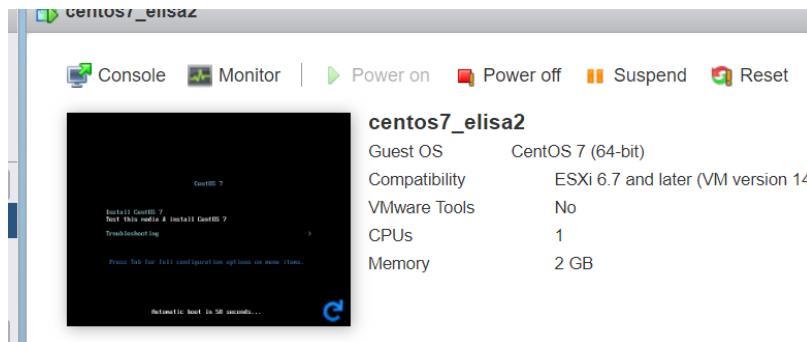
Una vez que tenemos la máquina virtual con el sistema operativo asociado, tenemos que iniciarla y configurarla correctamente para integrarla en nuestra red.

Name	Status	Size	Type
centos7_elisa2	Powered off	16 GB	Red Hat Enterprise Li...
centos7_Artemis_2016_01_01	Powered off	8.27 GB	CentOS 4/5 or later (...)
elisa2	Powered off	16 GB	CentOS 7 (64-bit)
elisa2_2016_01_01	Powered off	0 B	CentOS 7 (64-bit)
elisa2_2016_01_02	Powered off	14.63 GB	CentOS 4/5 or later (...)
elisa2_2016_01_03	Powered off	4.93 GB	CentOS 7 (64-bit)
elisa2_2016_01_04	Powered off	20.31 GB	CentOS 7 (64-bit)

### Tarea 1.3 - Iniciar y configurar la máquina virtual.

Para encender nuestra máquina virtual, tenemos que hacer clic en "Encender".

Una vez encendido, hacemos clic en la ventana negra que se nos muestra a continuación.



Y dentro de la ventana se ejecutará la instalación de CentOS 7.

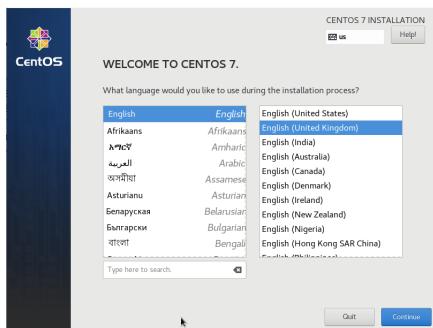
Además del usuario root, tendremos a otro usuario sin permisos de superusuario por temas de seguridad. Le daremos permisos exclusivamente a los recursos necesarios:

- root, contraseña: Elisa2astech
- elisauser, contraseña: Elisa2astech

Una vez que se carga el instalador, podemos ver el primer paso que es configurar el idioma y la región. Seleccione los correctos. Hacemos clic en "Continuar".

Después de la confirmación, observamos que el instalador necesita ser configurado.

Lo que hay



que

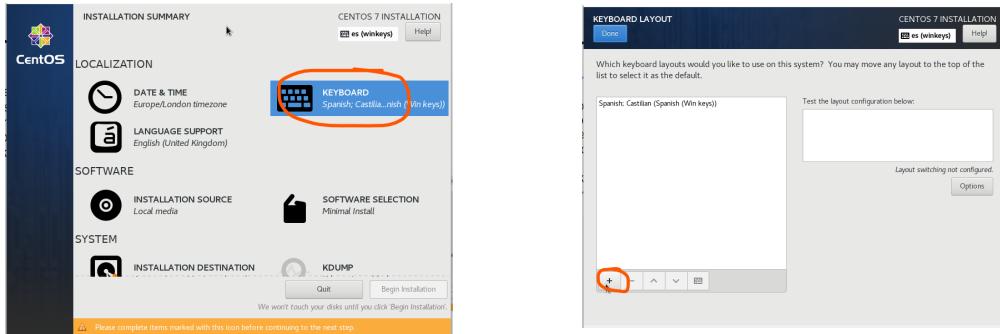


primero

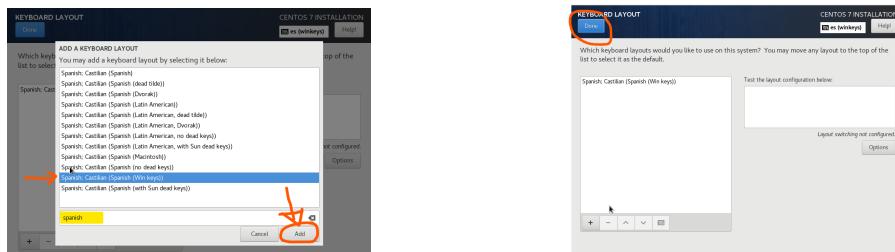
configurar es la distribución del teclado. Es importante utilizar el diseño adecuado de

nuestro teclado porque algunas de las teclas son diferentes según la distribución elegida de fábrica. En este caso, podemos observar que necesitamos el diseño español.

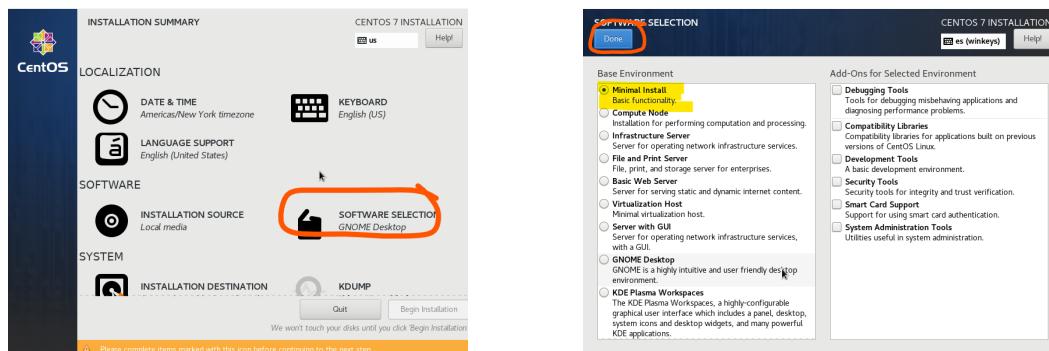
Hacemos clic en "Teclado" y luego en el botón "Agregar".



Escribimos el idioma deseado, elegimos el diseño y hacemos clic en "Agregar". Por último, presionamos “Listo”.

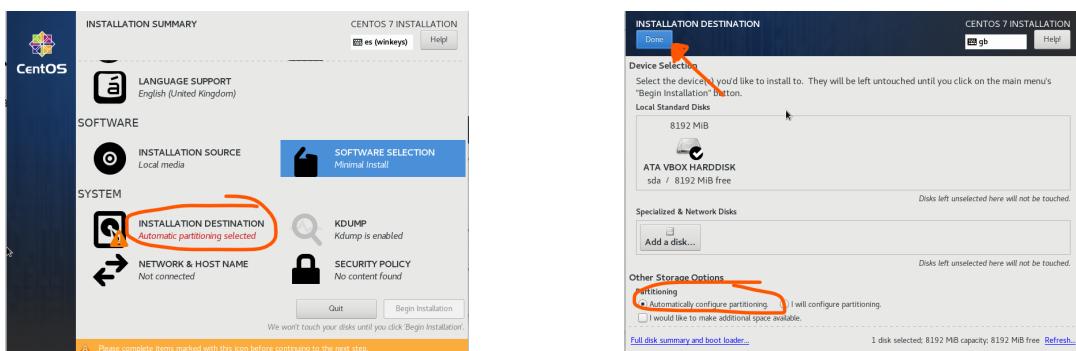


La selección de software será “Mínima”, porque vamos a instalar el software requerido más tarde mediante comandos.



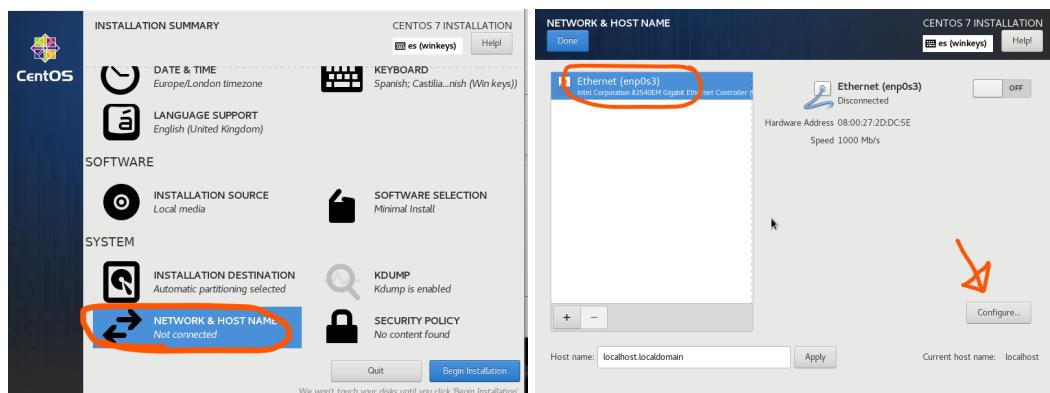
Hacemos clic en Instalación de software > “Listo”.

Fuente de instalación > "Configurar automáticamente la partición" > Hacemos clic en "Listo".



El sistema detectará el disco duro virtual creado con VMWare y asignará las particiones correspondientes al sistema Linux correspondiente. Es donde se alojarán todos nuestros ficheros y directorios base del sistema a instalar.

Toca configurar los ajustes de red y nombre de host. Primero, seleccione la interfaz en la lista. En este caso es Ethernet (emp0s3). Luego, haremos clic en "configurar".



Haga clic en "Configuración de IPv4". Luego, clic en "método" y lo cambiamos a "manual". Seleccionamos esta opción en el menú desplegable.

Haremos clic en “agregar” y escriba la información de la red que nos proporciona la empresa:

- Cualquier IP disponible en 192.xxx.x.0/xx o la que corresponda.
- Máscara de red: por ejemplo, 255.255.255.0 o la que corresponda.
- Puerta de enlace: 192.xxx.xxx.xxx o la que corresponda.



**Addresses**

Address	Netmask	Gateway	Add	Delete
192.168.1.110	255.255.255.0	192.168.1.254		

Address	Netmask	Gateway
192.168.1.110	255.255.255.0	192.168.1.254

Los servidores DNS se agregarán separados por “,”. Si todo es correcto, el siguiente paso es hacer clic en "Guardar".

→

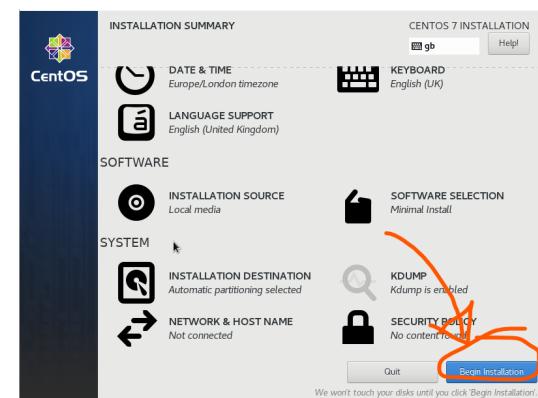
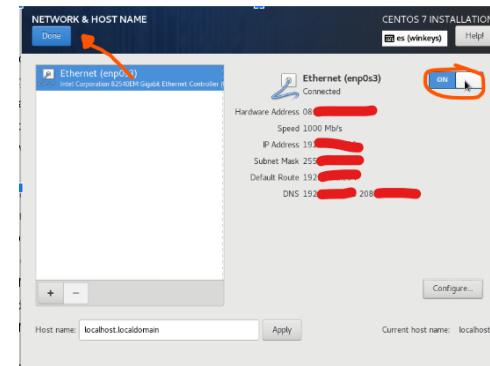
DNS servers:
192.168.1.253,208.60.220.220

El último paso para configurar la conexión es encender Ethernet. Luego, hacemos clic en "Listo".

Una vez realizado tendremos conectada nuestra máquina en red. Será identificable y accesible desde un navegador o programa de SSH.

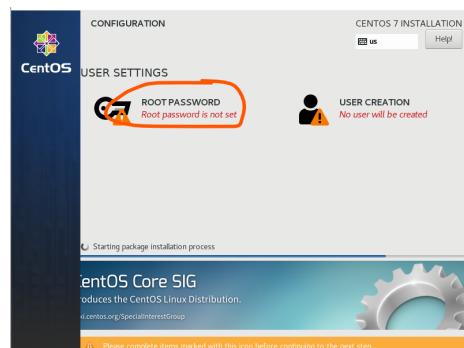
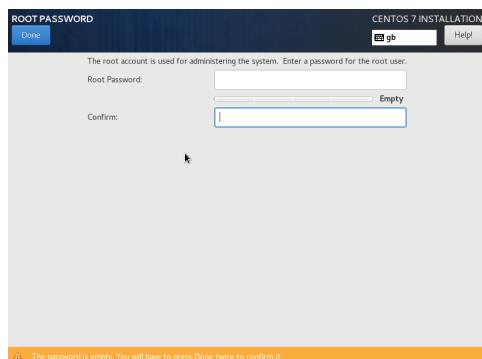
Finalizados todos estos pasos, se podrá iniciar la instalación.

Esta, dependiendo del sistema y los recursos disponibles, tardará de 5 a 15 minutos, aunque en máquinas modernas, suele ser una instalación sencilla y rápida, debido a la selección mínima de componentes. Los que necesitemos más adelante, los administraremos de forma manual mediante línea de comandos. De esa manera, tendremos una máquina ligera y escalable en un futuro.



En la siguiente pantalla configuraremos tanto a "root" como al usuario estándar, mientras se completa la instalación de los componentes básicos del sistema.

Configuración de la contraseña de root > Escriba la contraseña que desea configurar >  
Hacemos doble clic en "Listo".

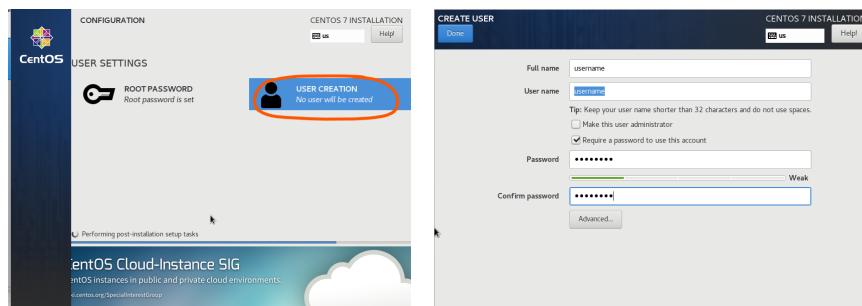


El sistema nos irá avisando de la seguridad de nuestra contraseña. Se considera una contraseña debe (al menos):

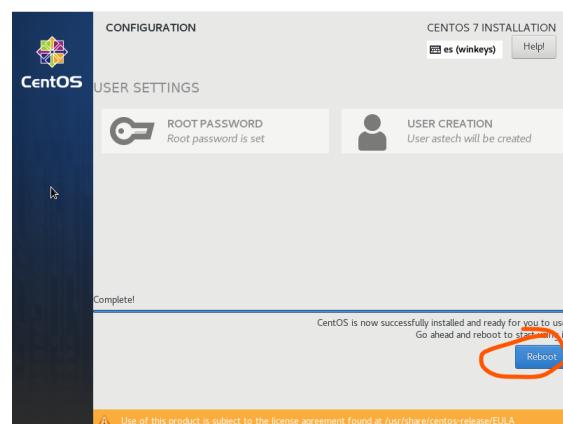
- Constar de 8 a 15 caracteres (mínimo).
- Utilizar MAYÚSCULAS y minúsculas.
- Utilizar caracteres especiales.
- Utilizar números de 0-9.

Lo mismo haremos con el usuario de sistema estándar:

Creación de usuario > Configuración de un usuario normal y una contraseña de usuario >  
Hacemos doble clic en "Listo".



Cuando finalice la instalación, aparecerá un botón "Reiniciar". Hacemos clic en él.



El sistema hará un reinicio para poder terminar de realizar las instalaciones,

comprobar que todo esté correctamente instalado y hará su primer inicio.

El primer arranque será algo como esto:

---

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

localhost login: _
```

Para iniciar sesión, escribimos el usuario que queremos usar y luego, la contraseña.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

localhost login: astech
Password:
[astech@localhost ~]#
```

Para iniciar sesión como root, debemos escribir "cerrar sesión" tras haber iniciado sesión con otro usuario. Escribimos "root", presionamos Intro y luego la contraseña.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

localhost login: root
Password:
[root@localhost ~]#
```

Ahora toca ver si Internet funciona correctamente. Hacemos un ping a los servidores de Google con *ping 8.8.8.8*.

**Password:**

```
[root@localhost ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=16.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=15.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=16.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=15.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=16.5 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 15.703/16.039/16.511/0.323 ms
[root@localhost ~]# _
```

Si tenemos conexión, podemos actualizar los paquetes del sistema usando:

*yum -y update*

```
[root@localhost ~]# yum -y update
Loaded plugins: fastestmirror
```

(55/110): kbd-legacy-1.15.5-16.el7_9.noarch.rpm	466 kB 00:00:00
(56/110): kbd-misc-1.15.5-16.el7_9.noarch.rpm	1.4 MB 00:00:00
(57/110): iwl7260-firmware-25.30.13.0-80.el7_9.noarch.rpm	6.1 MB 00:00:01
(58/110): kernel-tools-3.10.0-1160.59.1.el7.x86_64.rpm	8.2 MB 00:00:02
(59/110): kernel-tools-libs-3.10.0-1160.59.1.el7.x86_64.rpm	8.1 MB 00:00:01
(60/110): kexec-tools-2.0.15-51.el7_9.3.x86_64.rpm	351 kB 00:00:00
(61/110): kpartx-0.4.9-135.el7_9.x86_64.rpm	81 kB 00:00:00
(62/110): krb5-libs-1.15.1-51.el7_9.x86_64.rpm	809 kB 00:00:00
(63/110): libblkid-2.23.2-65.el7_9.1.x86_64.rpm	183 kB 00:00:00
(64/110): libcroco-0.6.12-6.el7_9.x86_64.rpm	105 kB 00:00:00
(65/110): libcurl-7.29.0-59.el7_9.1.x86_64.rpm	223 kB 00:00:00
(66/110): libmount-2.23.2-65.el7_9.1.x86_64.rpm	185 kB 00:00:00
(67/110): libsmartcols-2.23.2-65.el7_9.1.x86_64.rpm	143 kB 00:00:00
(68/110): libuuid-2.23.2-65.el7_9.1.x86_64.rpm	84 kB 00:00:00
(69/110): libxml2-2.9.1-6.el7_9.6.x86_64.rpm	668 kB 00:00:00
(70/110): libxmlsec-1.2.10-8.el7.x86_64.rpm	59 kB 00:00:00

Podemos apagar la máquina virtual con *shutdown -P 0* o *shutdown NOW*

### Tarea 1.4 - Configuración de la pila LAMP.

Una pila "LAMP" es un grupo de software de código abierto que normalmente se instalan juntos para permitir que un servidor aloje sitios web y aplicaciones web dinámicos. Este término es en realidad un acrónimo que representa el sistema operativo Linux, con el servidor web Apache. Los datos del sitio se almacenan en una base de datos MySQL (usando MariaDB) y PHP procesa el contenido dinámico.

- Linux: CentOS 7.
- Apache 2.
- Maria DB (MySQL).
- PHP.

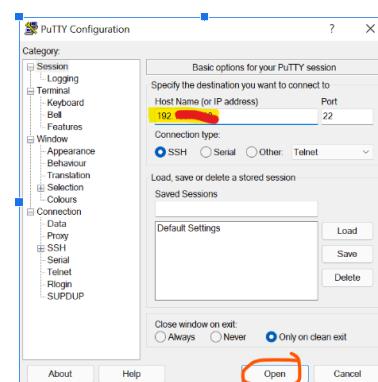
Antes de instalar nada, debemos actualizar los binarios del sistema operativo:

- Iniciamos sesión como usuario raíz.
- Escribimos el siguiente comando: *yum -y update*

```
localhost login: root
Password:
Last login: Tue Mar 29 13:05:34 on ttym1
[root@localhost ~]# yum -y update
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.strencom.net
```

Una vez realizado, en el sistema host donde gestionamos nuestra máquina virtual, vamos a instalar un programa que nos facilitará la gestión de la misma, PuTTY.

La interfaz tiene un conjunto de parámetros. En el Nombre del host, escribimos la dirección IP de la máquina virtual. El tipo de conexión será SSH. El siguiente paso es hacer clic en "Abrir".



Cada vez que queramos usar la máquina virtual, podremos ejecutar este programa. Hará más fácil la gestión y administración.

Simplemente iniciamos con el usuario deseado.

```
root@localhost:~  
login as: root  
root@192.168.1.10's password:  
Last login: Thu Mar 31 15:08:25 2022 from 192.168.1.10  
[root@localhost ~]#
```

Otra alternativa es Terminal de Windows, que viene instalada en Windows 11 por defecto e integra funcionalidades de SSH similares.

```
root@elisa2vm:~  
Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.  
Instale la versión más reciente de PowerShell para obtener nuevas caract  
PS C:\Users\andre> ssh root@192.168.1.10  
PS C:\Users\andre> ssh root@192.168.1.10  
root@192.168.1.10's password:  
Last login: Sun Jun 12 09:40:18 2022 from 192.168.1.10  
[root@elisa2vm ~]#
```

Conexión denegada:

```
PS C:\Users\andre> ssh root@192.168.1.10  
ssh: connect to host 192.168.1.10 port 22: Connection refused
```

Es un error común en las distribuciones de Red Hat, como CentOS. Para solucionar este problema tienes que hacer lo siguiente:

```
[root@elisa2vm ~]# sshd -t  
/var/empty/sshd must be owned by root and not group or world-writable.  
[root@elisa2vm ~]# chmod 755 /var/empty/sshd  
[root@elisa2vm ~]# service sshd restart  
Redirecting to /bin/systemctl restart sshd.service
```

*sshd-t*

Este comando hace una prueba y muestra el problema que tiene la aplicación. En este caso, es un problema con los permisos. Le damos permisos con el comando chmod:

*chmod 755 /var/empty/sshd*

Reiniciamos el servicio ssh:

*service sshd restart*

E intentamos acceder de nuevo con nuestro programa ssh preferido.

```
PS C:\Users\andre> ssh root@192.168.1.18
root@192.168.1.18's password:
Last login: Fri May  6 10:40:53 2022
```

### Tarea 1.4.1 -Instalación de Apache 2.4.6.

Apache HTTP Server es un software de servidor web multiplataforma gratuito y de código abierto, publicado bajo los términos de la Licencia Apache 2.0. Apache es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo los términos de Apache Software Foundation. El servidor web Apache es actualmente el servidor web más popular del mundo.

Para instalarlo debemos introducir el comando: *yum install -y httpd*

```
localhost login: root
Password:
Last login: Mon Mar 28 16:41:07 on tty1
[root@localhost ~]# yum install -y httpd_
[root@localhost ~]# yum install -y httpd_
Installed:
  httpd.x86_64 0:2.4.6-97.el7.centos.5
Dependency Installed:
  apr.x86_64 0:1.4.8-7.el7
  httpd-tools.x86_64 0:2.4.6-97.el7.centos.5
  mailcap.noarch 0:2.1.41-2.el7
  apr-util.x86_64 0:1.5.2-6.el7
```

Una vez que se instala, podemos iniciar Apache en el sistema:

```
systemctl start httpd.service
[root@localhost ~]# systemctl start httpd.service
[root@localhost ~]#
```

Finalmente, configure Apache para que comience en el arranque:

```
systemctl enable httpd.service
[root@localhost ~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
```

Para ver la versión actual de Apache:

```
httpd -v
```

```
[root@localhost ~]# httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:  Mar 24 2022 14:57:57
```

Apache está listo para usar, pero no podemos probar el funcionamiento porque necesitamos habilitarlo en el firewall. FirewallD está habilitado para CentOS 7, pero HTTP no está incluido en el conjunto predeterminado de servicios.

Con este comando podemos ver el conjunto predeterminado de servicios:

```
firewall-cmd --zone=public  
--list-services
```

```
[root@localhost ~]# firewall-cmd --zone=public --list-services  
dhcpv6-client ssh
```

Para permitir conexiones a Apache, agregamos a HTTP como un servicio:

```
firewall-cmd --zone=public --add-service=http
```

Podemos hacer que nuestro cambio sea permanente escribiendo:

```
firewall-cmd --zone=public --add-service=http --permanent
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-service=http --permanent  
success  
[root@localhost ~]# firewall-cmd --zone=public --add-service=http  
success
```

Apache ahora permite el tráfico web HTTP en el puerto 80. Si queremos usar SSL/TLS, es decir, conexión segura, también debemos agregar el servicio HTTPS. Podemos agregar eso a la sesión actual y al conjunto de reglas permanentes escribiendo:

```
firewall-cmd --zone=public --add-service=https
```

```
firewall-cmd --zone=public --add-service=https --permanent
```

Comprobamos de nuevo los cambios con `firewall-cmd --zone=public --list-services`:

```
[root@localhost ~]# firewall-cmd --zone=public --list-services  
dhcpv6-client http https ssh
```

Después de aplicar permanentemente estas reglas, se puede reiniciar la red y recargar el servicio de firewall:

```
systemctl restart network
```

```
systemctl reload firewalld
```

Y lo mismo con Apache:

```
systemctl reload httpd.service
```

```
systemctl restart httpd.service
```

```
[root@localhost ~]# systemctl reload httpd.service
[root@localhost ~]# systemctl restart httpd.service
[root@localhost ~]# systemctl restart network
[root@localhost ~]# systemctl reload firewalld
[root@localhost ~]# systemctl reload httpd.service
[root@localhost ~]# systemctl restart httpd.service
```

Lo siguiente que haremos es comprobar la ip de nuestro servidor con *ip -a*:

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:2d:dc:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.24 brd 192.168.1.24 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::dab3:b3de:ecdb:d411/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Podemos hacer una verificación para verificar que todo salió según lo planeado visitando la dirección IP pública del servidor en un navegador web:

[http://vm\\_ip](http://vm_ip)

Por ejemplo: <http://192.xxx.xxx.xxx>

Verá la página web predeterminada de CentOS 7 Apache, que está para fines informativos y de prueba. Debería verse algo como esto:



Apache está listo y configurado correctamente.

## Tarea 1.4.2 -Instalación de MySQL (MariaDB) 10.4.24.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius, fundador de MySQL, la fundación MariaDB y la comunidad de desarrolladores de software libre. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Hay que comprobar que estamos ejecutando un CentOS 7 actualizado con *yum -y update*.

Después de actualizar el sistema, agregue el repositorio MariaDB en YUM, el instalador de paquetes por defecto de CentOS:

```
tee /etc/yum.repos.d/MariaDB.repo<<EOF
[mariadb]
name = MariaDB
baseurl =
http://yum.mariadb.org/10.4/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
EOF
```

```
[root@localhost ~]# tee /etc/yum.repos.d/MariaDB.repo<<EOF
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.4/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
EOF
```

Actualizamos el índice de caché de yum:

```
yum makecache fast
```

```
[root@localhost ~]# yum clean all
Loaded plugins: fastestmirror
Cleaning repos: base extras mariadb updates
Cleaning up list of fastest mirrors
[root@localhost ~]#
```

Tenemos que tener en cuenta que si se cambia la configuración de un repositorio existente, se debe ejecutar lo siguiente: *yum clean all*

```
[root@localhost ~]# yum makecache fast
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.strencom.net
 * extras: mirror.strencom.net
 * updates: mirror.strencom.net
base
extras
mariadb
updates
Metadata Cache Created
```

Ahora es momento de instalar los paquetes más comunes, con el siguiente comando:

```
yum -y install MariaDB-server galera-4 MariaDB-client MariaDB-shared MariaDB-backup
MariaDB-common
```

```
[root@localhost ~]# yum install -y MariaDB-server galera-4 MariaDB-client MariaDB-shared MariaDB-backup
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirror.strencom.net
 * extras: mirror.strencom.net
 * updates: mirror.strencom.net
base
extras
mariadb
updates
(1/6): base/7/x86_64/group_gz | 3.6 kB 00:00:00
(2/6): mariadb/updateinfo | 2.9 kB 00:00:00
(3/6): mariadb-primary_db | 3.4 kB 00:00:00
(4/6): extras/7/x86_64/primary_db | 2.9 kB 00:00:00
(5/6): base/7/x86_64/mariadb |
153 kB 00:00:00
5.8 kB 00:00:00
63 kB 00:00:00
246 kB 00:00:00
6.1 MB 00:00:02
```

Cuando se complete la instalación, debemos iniciar MariaDB con el comando:  
`systemctl start mariadb`

Ahora es el momento de ejecutar un script de seguridad simple que eliminará algunos valores predeterminados peligrosos y bloqueará el acceso a nuestro sistema de base de datos ejecutando este script:

`mysql_secure_installation`

Vamos a seleccionar los valores del siguiente modo:

*Switch to unix\_socket authentication: No (we don't want to use unix users for login)*

*The prompt will ask you for the current root password. Since we just installed MySQL, we won't have one, so leave it blank by pressing enter.*

*Then the prompt will ask you if you want to set a root password. Set a secure one.*

*Remove anonymous users: y*

*Disallow root login remotely: y (we can set up this option later to work remotely)*

*Remove test database and access to it: y*

*Reload privilege tables now?: y*

Cuando terminemos, debemos habilitar MariaDB en el arranque:

`systemctl enable mariadb.service`

```
[root@localhost ~]# systemctl enable mariadb.service
Created symlink from /etc/systemd/system/mysql.service to /usr/lib/systemd/system/mariadb.service.
Created symlink from /etc/systemd/system/mysqld.service to /usr/lib/systemd/system/mariadb.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
```

Podemos ejecutar mysql con este comando:

```
mysql -user root -password o mysql -u root -p
```

To exit mysql:

*exit;*

Tendríamos el gestor de bases de datos listo para usar.

### **Tarea 1.4.3 -Instalación de PHP 8.1.4.**

PHP son las siglas en inglés de “Hypertext Pre-Processor”. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Se utiliza para crear nuestras aplicaciones web y páginas web dinámicas.

Dado que los repositorios YUM predeterminados en CentOS 7 no contienen paquetes PHP 8.1 (solo PHP 5.4 que está desactualizado), usaremos el repositorio REMI para instalar 8.1 en CentOS 7.

Pero en primera instancia necesitamos instalar yum-utils, que es una colección de herramientas y programas para administrar repositorios yum, instalar paquetes de depuración, paquetes fuente, información extendida de repositorios y administración:

```
yum -y update && yum install -y yum-utils
[root@localhost ~]# yum -y update && yum install -y yum-utils

Installed:
  yum-utils.noarch 0:1.1.31-54.el7_8

Dependency Installed:
  libxml2-python.x86_64 0:2.9.1-6.el7_9.6
  python-kitchen.noarch 0:1.1.1-5.el7

Complete!
```

El siguiente paso es instalar Extra Packages for Enterprise Linux (EPEL), que nos permite ampliar nuestros paquetes y mantenerlos actualizados.

```
yum install -y epel-release
```

```
[root@localhost ~]# yum install -y epel-release_
```

```
Installed:  
epel-release.noarch 0:7-11
```

```
Complete!
```

A continuación, necesitaremos el repositorio Remi RPM, que es un repositorio YUM gratuito y estable principalmente para la pila de PHP. Contiene paquetes para las últimas versiones de PHP.

```
yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

```
[root@localhost ~]# yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

```
Installed:  
remi-release.noarch 0:7.9-3.el7.remi
```

```
Complete!
```

Luego, habilitamos el módulo PHP 8.1.

```
yum-config-manager --enable remi-php81
```

```
[root@localhost ~]# yum-config-manager --enable remi-php81_
```

Y luego, actualice los repositorios nuevamente:

```
yum -y update
```

```
mirrorlist_expire = 06480
name = Remi's PHP 8.1 RPM repository for Enterprise Linux 7 - x86_64
old_base_cache_dir =
password =
persistdir = /var/lib/yum/repos/x86_64/7/remi-php81
pkkdir = /var/cache/yum/x86_64/7/remi-php81/packages
proxy = False
proxy_dict =
proxy_password =
proxy_username =
repo_gpgcheck = False
retries = 10
skip_if_unavailable = False
ssl_check_cert_permissions = True
sslcacert =
sslcacert =
sslcacertkey =
sslvverify = True
throttle = 0
timeout = 30.0
ui_id = remi-php81
ui_repolist_vars = releasever,
basearch
username =
```

Y finalmente, después de habilitar el módulo PHP 8.1, podemos instalar PHP 8.1 con el siguiente comando:

```
yum install -y php
```

Podemos verificar si se instaló PHP ejecutando: `php -v`

```
[root@localhost ~]# php -v
PHP 8.1.4 (cli) (built: Mar 15 2022 20:18:41) (NTS gcc x86_64)
Copyright (c) The PHP Group
Zend Engine v4.1.4 Copyright (c) Zend Technologies
```

```
[root@localhost ~]# yum install -y php_
```

Installed:

```
php.x86_64 0:8.1.4-1.el7.remi
```

Dependency Installed:

```
libsodium.x86_64 0:1.0.18-1.el7
```

```
php-common.x86_64 0:8.1.4-1.el7.remi
```

Complete!

Ahora le toca el turno a las extensiones de PHP más comunes, para evitar futuras incompatibilidades con cierto tipos de usos.

```
yum -y install php-fpm php-curl php-cli
php-json php-mysql php-opcache php-dom
php-exif php-fileinfo php-zip php-mbstring
php-hash php-imagick php-openssl
php-pcre php-xml php-bcmath php-filter
php-pear php-gd php-mcrypt php-intl
php-iconv php-zlib php-xmlreader
```

```
[root@localhost ~]# yum install -y php-fpm php-curl php-cgi php-json php-xml php-opcache php-dom
php-exif php-fileinfo php-zip php-mbstring php-hash php-imagick php-openssl php-pcre php-xml php-bcmath
php-filter php-pear php-gd php-mcrypt php-intl php-iconv php-zlib php-xmlreader
urw-base35-d050001-fonts.noarch 0:20170801-10.el7
urw-base35-fonts.noarch 0:20170801-10.el7
urw-base35-fonts-common.noarch 0:20170801-10.el7
urw-base35-gothic-fonts.noarch 0:20170801-10.el7
urw-base35-nimbus-mono-ps-fonts.noarch 0:20170801-10.el7
urw-base35-nimbus-roman-fonts.noarch 0:20170801-10.el7
urw-base35-nimbus-sans-fonts.noarch 0:20170801-10.el7
urw-base35-p052-fonts.noarch 0:20170801-10.el7
urw-base35-standard-symbols-ps-fonts.noarch 0:20170801-10.el7
urw-base35-z003-fonts.noarch 0:20170801-10.el7
xorg-x11-font-utils.x86_64 1:7.5-21.el7
xorg-x11-server-utils.x86_64 0:7.7-20.el7
Complete!
```

Y ya tendríamos completa nuestra instalación de PHP. Por último, realizaremos una prueba del mismo en el navegador.

Para probar que nuestro sistema está configurado correctamente para PHP, podemos crear un script PHP muy básico.

Llamaremos a este script `info.php`. Para que Apache encuentre el fichero y lo sirva correctamente, debe guardarse en un directorio muy específico, que se denomina "raíz web". En CentOS 7, este directorio se encuentra en `/var/www/html/`. Podemos crear el fichero en esa ubicación escribiendo:

```
vi /var/www/html/info.php
```

```
[root@localhost ~]# vi /var/www/html/info.php
```

Esto abrirá un fichero en blanco. Presiona "a" para editar con el siguiente texto, que es un código PHP válido, dentro del archivo:

```
<?php phpinfo(); ?>
```

```
<?php phpinfo(); ?>
```

“ESC” + “:wq” para guardar y cerrar el fichero.

Una vez hecho, debemos reiniciar Apache con el comando `systemctl restart httpd`

Dentro de un navegador, escribirmos nuestra dirección local: <http://192.xxx.xxx.xxx> por ejemplo.

Este fichero deberá de ser eliminado por razones de seguridad.

En este punto, tendremos PHP listo para ejecutarse cuando se requiera.

#### PHP Version 8.1.4

System	Linux localhost.localdomain
Build Date	Mar 15 2022 20:18:41
Build System	Red Hat Enterprise Linux
Build Provider	Remi's RPM repository <https://rpms.remirepo.net/>
Compiler	gcc (GCC) 10.2.1 20210105 (Red Hat 10.2.1-1)

#### Tarea 1.4.4 -Instalación de phpMyAdmin 5.1.3-2.el7.remi (opcional).

PhpMyAdmin es una aplicación web que se utiliza para gestionar bases de datos MySQL de forma sencilla y con una interfaz amigable. Es un software muy popular basado en PHP. La ventaja de utilizar una aplicación web es que nos permite conectarnos a servidores remotos, a los que no siempre se puede acceder mediante programas de interfaz gráfica.

Puede ser un riesgo para la seguridad, ya que si no se configura correctamente, esta aplicación puede ser aprovechada por atacantes.

También está sujeto al soporte de su desarrollador, en cualquier caso, si tiene problemas de seguridad, depende del desarrollador solucionarlos.

Puede ser una herramienta útil en entornos de desarrollo no expuestos a producción.

Para instalar PHPMyAdmin en CentOS, ingresamos el comando:

```
yum -y install phpmyadmin
```

Hay que reiniciar Apache con:

```
systemctl restart httpd.service
```

Una vez que el sistema ejecuta el comando, PHPMyAdmin está instalado y listo para ejecutarse.

```
Installed:
  phpMyAdmin.noarch 0:4.4.15.10-6.el7

Dependency Installed:
  libtidy.x86_64 0:5.4.0-1.el7
  php-fedorautoloader.noarch 0:1.0.1
  php-php-gettext.noarch 0:1.0.12-1.el7
  php-tcpdf.noarch 0:6.2.26-1.el7
  php-tcpdf-dejavu-sans-fonts.noarch 0
  php-tidy.x86_64 0:8.1.4-1.el7.remi

Complete!
```

De forma predeterminada, solo se puede acceder a phpMyAdmin desde el servidor local. Entonces, si intentamos conectarnos a phpMyAdmin con [http://public\\_IP/phpmyadmin](http://public_IP/phpmyadmin), el navegador devolverá un 403 Forbidden, es decir, no estamos autorizados a acceder. Para acceder a phpMyAdmin desde máquinas externas, debemos editar el archivo *phpMyAdmin.conf* con el comando:

```
vi /etc/httpd/conf.d/phpMyAdmin.conf
```

En CentOS 7, el acceso web es administrado por el módulo *mod\_authz\_core.c*. Por lo tanto, las reglas de permitir o denegar no funcionarán incluso si las modifica.

La configuración predeterminada se verá así →

```
<Directory /usr/share/phpMyAdmin/>
    AddDefaultCharset UTF-8
<IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
        Require ip 127.0.0.1
        Require ip ::1
    </RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
</IfModule>
</Directory>
```

```
<Directory /usr/share/phpMyAdmin/setup/>
<IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
        Require ip 127.0.0.1
        Require ip ::1
    </RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
</IfModule>
</Directory>
```

Comentamos *Require ip 127.0.0.1* y *Require ip ::1* y luego agregue *Require all granted* justo debajo de la línea comentada. El contenido se verá como a continuación después de la modificación. "a" para editar y "Esc" + ":wq" para guardar y salir.

Una vez que lo hicimos, reiniciamos Apache:  
`systemctl restart httpd`

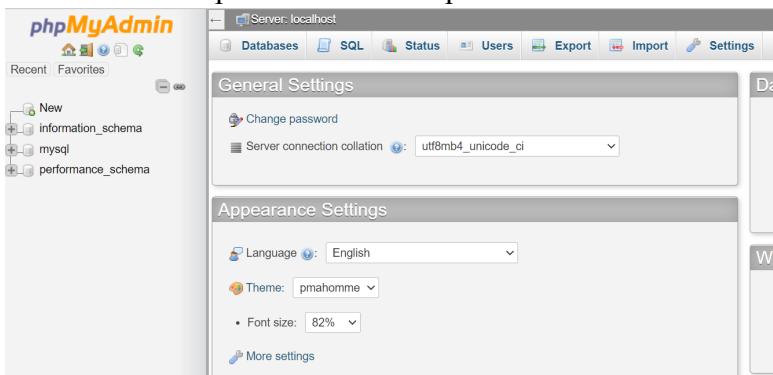
Ahora ponemos [http://local\\_ip/phpmyadmin](http://local_ip/phpmyadmin) en el navegador e iniciamos sesión con root y la contraseña asignada:



```
<IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
        # Require ip 127.0.0.1
        # Require ip ::1
        Require all granted
    </RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
</IfModule>
</Directory>
```

```
<Directory /usr/share/phpMyAdmin/setup/>
<IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
        # Require ip 127.0.0.1
        # Require ip ::1
        Require all granted
    </RequireAny>
</IfModule>
```

Y este sería el aspecto de nuestra aplicación nada más iniciar:



Si vemos en la página principal, en la esquina inferior derecha, es phpMyAdmin 4.4.15.10, que está desactualizado. Queremos phpMyAdmin 5.

Entonces necesitamos habilitar el repositorio EPEL/Remi que instalamos antes.

```
yum --enablerepo=remi install  
phpmyadmin -y
```

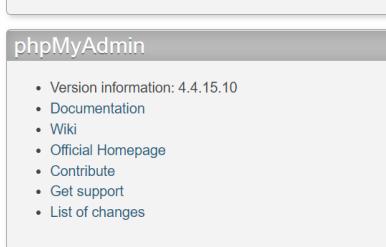
Updated:

```
phpMyAdmin.noarch 0:5.1.3-2.el7.remi
```

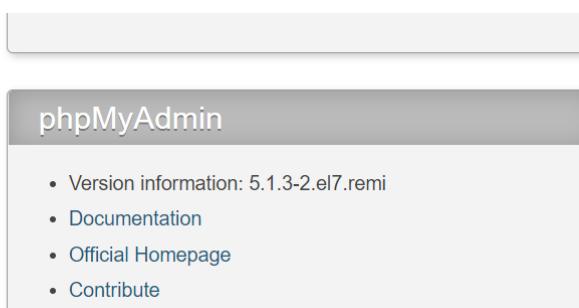
Reiniciamos apache:

Complete!

```
systemctl restart httpd
```



E ingresamos de nuevo a la aplicación. Se observa que ahora está en su última versión.



### Tarea 1.4.5 - Configuración de SELINUX y Firewall.

Security-Enhanced Linux (SELinux) es una arquitectura de seguridad para sistemas Linux que permite a los administradores tener más control sobre quién puede acceder al sistema. Fue desarrollado originalmente por la Agencia de Seguridad Nacional de los Estados Unidos (NSA) como una serie de parches para el kernel de Linux utilizando Módulos de Seguridad de Linux (LSM).

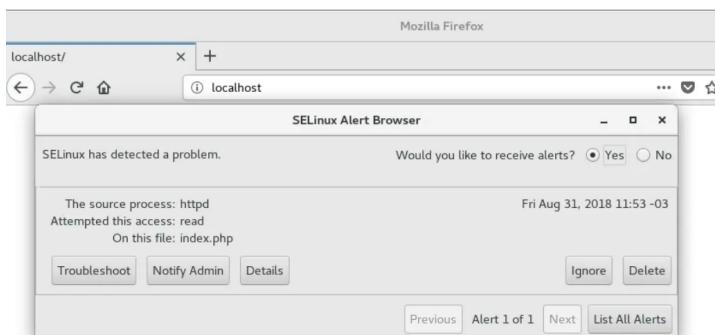
SELinux se lanzó a la comunidad de código abierto en 2000 y se integró en el kernel de Linux ascendente en 2003.

SELinux define controles de acceso para las aplicaciones, procesos y archivos en un sistema. Utiliza políticas de seguridad, que son un conjunto de reglas que le dicen a SELinux a qué se puede o no acceder, para hacer cumplir el acceso permitido por una política.

SELinux viene habilitado por defecto en CentOS 7 y su configuración por defecto es restringir el acceso de Apache a los directorios hasta que se otorguen permisos explícitos o como hacen algunos y ya hemos visto, lo deshabilitan lo cual no es recomendable.

Si no hacemos esto nos podemos encontrar con un mensaje como:

- Navegador de alertas de SELinux
- SELinux ha detectado un problema.
- El proceso fuente: httpd



Usaremos *chown* para hacer que Apache sea propietario del directorio web:

*chown apache:apache -R /var/www/html/*

[root@localhost ~]# chown apache:apache -R /var/www/html/

Modificaremos permisos para ficheros y directorios:

*cd /var/www/html/*

```
[root@localhost ~]# cd /var/www/html/  
[root@localhost html]# find . -type f -exec sudo chmod 0644 {} \;  
[root@localhost html]# find . -type d -exec sudo chmod 0755 {} \;
```

*find . -type f -exec sudo chmod 0644 {} \;  
find . -type d -exec sudo chmod 0755 {} \;*

Usaremos *chcon* para cambiar el contexto de seguridad del archivo para el contenido web:

```
chcon -t httpd_sys_content_t /var/www/html/ -R
```

```
chcon -t httpd_sys_rw_content_t /var/www/html/ -R
```

```
[root@localhost html]# chcon -t httpd_sys_content_t /var/www/html/ -R
[root@localhost html]# chcon -t httpd_sys_rw_content_t /var/www/html/ -R
```

Reiniciamos el servicio para que se realicen los cambios anteriores:

```
systemctl restart httpd.service
```

```
[root@localhost html]# systemctl restart httpd.service
```

Ahora toca comprobar los ajustes de Firewall, ya que, en conjunto con SELINUX, aplican restricciones a los parámetros de nuestro sistema.

FirewallD está habilitado para CentOS 7, pero HTTP no está incluido en el conjunto predeterminado de servicios.

Ver el conjunto predeterminado de servicios:

```
firewall-cmd --zone=public --list-services
```

```
[root@localhost ~]# firewall-cmd --zone=public --list-services
dhcpv6-client ssh
```

Para permitir conexiones a Apache, agregamos HTTP y HTTPS como un servicio:

```
firewall-cmd --zone=public --add-service=http
```

```
firewall-cmd --zone=public --add-service=https
```

Y hacemos estos cambios permanentes:

```
firewall-cmd --zone=public --add-service=http --permanent
```

```
firewall-cmd --zone=public --add-service=https --permanent
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-service=http --permanent
success
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-service=https --permanent
success
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-service=https --permanent
success
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-service=https --permanent
success
```

Y volvemos a comprobar el conjunto predeterminado de servicios:  
`firewall-cmd --zone=public --list-services`

```
[root@localhost ~]# firewall-cmd --zone=public --list-services
dhcpv6-client http https ssh
```

Después de aplicar permanentemente estas reglas, reiniciamos la red y el servicio de firewall:

```
systemctl restart network
systemctl reload firewalld
```

Y lo mismo con Apache:

```
systemctl reload httpd.service
systemctl restart httpd.service
```

```
[root@localhost ~]# systemctl reload httpd.service
[root@localhost ~]# systemctl restart httpd.service
[root@localhost ~]# systemctl restart network
[root@localhost ~]# systemctl reload firewalld
[root@localhost ~]# systemctl reload httpd.service
[root@localhost ~]# systemctl restart httpd.service
```

Comprobamos que Apache funcione correctamente poniendo nuestra IP en el navegador:

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:2d:dc:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.24 brd 192.168.1.255 scope global nopref ixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::dab3:b3de:ecdb:d411/64 scope link nopref ixroute
        valid_lft forever preferred_lft forever
```

# Testing 123..

This page is used to test the proper operation of the [Apache HTTP server](#) after it has been installed. If you can read this page it means that this site is working properly. This server is powered by [CentOS](#).

## Tarea 2 - Instalación de Laravel.

Laravel es un framework PHP de código abierto, que es robusto y fácil de entender. Sigue un patrón de diseño modelo-vista-controlador. Laravel reutiliza los componentes existentes de diferentes frameworks que ayudan a crear una aplicación web. La aplicación web así diseñada es más estructurada y pragmática.

Laravel ofrece un rico conjunto de funcionalidades que incorpora las características básicas de los marcos PHP como CodeIgniter, Yii y otros lenguajes de programación como Ruby on Rails. Laravel tiene un conjunto muy rico de características que aumentarán la velocidad del desarrollo web.

Laravel ofrece las siguientes ventajas, cuando se está diseñando una aplicación web basada en él:

- La aplicación web se vuelve más escalable gracias al framework Laravel.
- Se ahorra un tiempo considerable en el diseño de la aplicación web, ya que Laravel reutiliza los componentes de otro marco en el desarrollo de la aplicación web.
- Incluye espacios de nombres e interfaces, por lo que ayuda a organizar y administrar los recursos.

Ya está todo listo para el despliegue de aplicaciones Laravel, pero falta instalar paso a paso los componentes.

Composer es una herramienta que incluye todas las dependencias y bibliotecas. Permite a un usuario crear un proyecto con respecto al marco mencionado (por ejemplo, los utilizados en la instalación de Laravel). Las bibliotecas de terceros se pueden instalar fácilmente con la ayuda de Composer.

Vamos a un directorio en el que tengamos permisos de escritura y escribimos los comandos:  
*cd ~*

```
get composer  
curl -sS https://getcomposer.org/installer | php
```

```
[root@localhost ~]# curl -sS https://getcomposer.org/installer | php  
All settings correct for using Composer  
Downloading...
```

```
Composer (version 2.3.2) successfully installed to: /root/composer.phar  
Use it: php composer.phar
```

Mueva Composer a un directorio bin en el que se tenga control:

```
mv composer.phar /usr/local/bin/composer
```

Permitimos ejecutar el archivo como un programa:

```
chmod +x /usr/local/bin/composer
```

```
[root@localhost ~]# chmod +x /usr/local/bin/composer
```

Verificamos que Composer funciona:

```
composer about
```

Confirmamos escribiendo “yes”. Todo deberá de funcionar correctamente.

```
[root@localhost ~]# mv composer.phar /usr/local/bin/composer
[root@localhost ~]# composer about
Do not run Composer as root/super user! See https://getcomposer.org/doc/04-schema.md#details
Continue as root/super user [yes]? yes
Composer - Dependency Manager for PHP - version 2.3.2
Composer is a dependency manager tracking local dependencies and libraries.
See https://getcomposer.org/ for more information.
```

Podemos instalar la última versión de Laravel usando los siguientes comandos:

Nos movemos a Apache virtualhost:

```
cd /var/www/html
```

Y toca instalar Git, que es un software para rastrear cambios en cualquier conjunto de archivos, generalmente utilizado para coordinar el trabajo entre programadores que desarrollan código fuente en colaboración durante el desarrollo del software. Lo instalamos con:

```
yum install -y git
```

Y descargamos el proyecto Laravel oficial desde su Github:

```
git clone https://github.com/laravel/laravel.git
```

```
Installed:
  git.x86_64 0:1.8.3.1-23.el7_8

Dependency Installed:
  perl-Error.noarch 1:0.17020-2.el7
  perl-Git.noarch 0:1.8.3.1-23.el7_8
  perl-TermReadKey.x86_64 0:2.30-20.el7

Complete!
```

```
receiving objects: 98% (32465/33127), 1.96 M
Receiving objects: 99% (32796/33127), 7.96 M
remote: Total 33127 (delta 3), reused 9 (delta 9)
reused 33111
Receiving objects: 100% (33127/33127), 7.96 M
Receiving objects: 100% (33127/33127), 10.16 MiB/s, done.
Resolving deltas: 100% (19600/19600), done.
[root@localhost html]#
```

Después de introducir el comando anterior, usamos Composer para instalar los recursos descargados de Laravel:

```
cd /var/www/html/laravel
composer install
```

Y escribimos “yes”.

Composer se encargará de ejecutar y desempaquetar los fuentes y directorios necesarios para poder ejecutar Laravel correctamente.

Cambiemos el nombre del proyecto predeterminado de "laravel" a "elisa" con:

```
cd /var/www/html
mv laravel elisa
```

La instalación de requisitos previos es un poco larga, pero debemos esperar hasta que esté completamente instalada para luego ejecutar los siguientes comandos que cambian los permisos:

```
chown -R apache:apache /var/www/html/elisa
chmod -R 755 /var/www/html/elisa
chmod -R 755 /var/www/html/elisa/storage
```

```
[root@localhost laravel]# chown -R apache:apache /var/www/html/laravel
[root@localhost laravel]# chmod -R 755 /var/www/html/laravel
[root@localhost laravel]# chmod -R 755 /var/www/html/laravel/storage
[root@localhost laravel]# chcon -R -t httpd_sys_rw_content_t /var/www/html/laravel/storage
[root@localhost laravel]#
```

Usamos el comando para escribir en el directorio de almacenamiento:

```
chcon -R -t httpd_sys_rw_content_t /var/www/html/elisa/storage
```

Ahora agregamos un host virtual al archivo de configuración de Apache para acceder a Laravel a través de la navegación web:

```
vi /etc/httpd/conf.d/vhost.conf
```

```
[root@localhost laravel]# pwd
/var/www/html/laravel
[root@localhost laravel]# composer install
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Continue as root/super user [yes]? yes
```

```
> Illuminate\Foundation\ComposerScript
> @php artisan package:discover --ansi
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/colii
Discovered Package: spatie/laravel-ide-helper
Package manifest generated successfully.
78 packages you are using are looking for updates.
Use the `composer fund` command to find them.
> @php artisan vendor:publish --tag=laravel --force
No publishable resources for tag [laravel].
Publishing complete.
[root@localhost laravel]#
```

```
[root@localhost html]# ls
info.php laravel
[root@localhost html]# mv laravel elisa2
[root@localhost html]# ls
elisa2 info.php
```

Y luego, añadimos el siguiente código:

```
<VirtualHost *:80>
    ServerName 192.168.1.110
    DocumentRoot /var/www/html/elisa/public
    <Directory /var/www/html/elisa>
        AllowOverride All
    </Directory>
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName 192.168.1.110
    DocumentRoot /var/www/html/laravel/public
    <Directory /var/www/html/laravel>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Guardamos el archivo de configuración.

Ahora es el momento de configurar la clave de cifrado. Laravel usa el archivo .env para configurar el entorno. Se utiliza el archivo .env para configurar todas las variables de entorno de su programa, como base de datos, SMTP, clave de seguridad, etc.

```
cd elisa
cp .env.example .env
```

Ahora configure la clave de encriptación aleatoria de 32 bits proporcionada:

```
php artisan key:generate
```

```
[root@localhost laravel]# cp .env.example .env
[root@localhost laravel]# php artisan key:generate
Application key set successfully.
```

Editamos el archivo de configuración de Apache:

```
vi /etc/httpd/conf/httpd.conf
```

Vamos a navegar hasta <Directory "/var/www/html"> y establecemos AllowOverride en "ALL".

```
# Options FileInfo AuthConfig
# AllowOverride ALL
```

```
# Further relax access to the default document root:
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes FollowSymLinks SymLinksIfOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None
```

Finalmente, reiniciamos Apache: `systemctl restart httpd`

Abrimos el navegador y escriba `http://ip_pública`

Laravel se ejecutará correctamente en el servidor.

### Tarea 3 - Instalación y configuración de ProFTPD y Filezilla.

#### Tarea 3.1 - ProFTPD, SELinux, Firewall y Passive FTP.

FTP (Protocolo de transferencia de archivos) es probablemente el método más popular para cargar archivos en un servidor; existe una amplia gama de servidores FTP, como ProFTPD, y clientes para cada plataforma.

ProFTPD es parte de Extra Packages for Enterprise Linux (EPEL), que es un repositorio comunitario de paquetes no estándar para la distribución RHEL (Red Hat Enterprise Linux, distribución en la que se basa CentOS).

Primero, instalaremos el repositorio EPEL → `yum install -y epel-release`

Lo primero que debe hacer es una actualización de paquetes:  
`yum -y update`

Luego instalaremos ProFTPD y cualquier paquete requerido:  
`yum -y install proftpd proftpd-utils`

```
Installed:
  proftpd.x86_64 0:1.3.5e-11.el7
Dependency Installed:
  GeoIP.x86_64 0:1.5.0-14.el7
  libevent.x86_64 0:2.0.21-4.el7
Complete!
```

En primer lugar, necesitamos configurar el nombre de host:  
`hostnamectl set-hostname elisavm.local`

Con `hostname` comprobamos los cambios realizados:

```
[root@localhost html]# hostnamectl set-hostname elisa2vm.local
[root@localhost html]# hostname
elisa2vm.local
```

“/etc/hosts” es un archivo utilizado por el sistema operativo para traducir nombres de host a direcciones IP. Lo editamos con nuestra información:

```
vi /etc/hosts
192.xxx.xxx.xxx elisavm.local localhost.localdomain
```

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.██████████ elisa2vm.local localhost.localdomain|
```

Editemos el archivo de configuración para ProFTPD como en la imagen:

```
vi /etc/proftpd.conf
```

```
ServerName "elisa2vm.local"
ServerIdent on "FTP Server ready."
ServerAdmin root@localhost
DefaultServer on

# Cause every FTP user except adm to be chrooted into their home directory
# DefaultRoot ~ !adm
DefaultRoot /var/www/html/elisa2/ !adm

# Use pam to authenticate (default) and be authoritative
# AuthPAMConfig proftpd
# AuthOrder mod_auth_pam.c* mod_auth_unix.c
# If you use NIS/YP/LDAP you may need to disable PersistentPasswd
```

DefaultRoot enjaula a los usuarios en el directorio establecido para que sólo accedan de manera predeterminada al recurso especificado. Es de vital importancia dar permiso de acceso sólo a los recursos necesarios para evitar cualquier problema de seguridad.

Vuelva a cargar el archivo de configuración nuevo:  
**systemctl reload proftpd**

Reinic peace el servicio ProFTPD:  
**systemctl restart proftpd**

Luego configure el servicio ProFTPD para que comience en el arranque:

```
[root@astechvm ~]# systemctl reload proftpd
[root@astechvm ~]# systemctl restart proftpd
[root@astechvm ~]# systemctl enable proftpd
[root@astechvm ~]# systemctl status proftpd
● proftpd.service - ProFTPD FTP Server
   Loaded: loaded (/usr/lib/systemd/system/proftpd.service; enabled)
   Active: active (running) since Fri 2022-04-01 14:20:28 BS
     Main PID: 1579 (proftpd)
        CGroup: /system.slice/proftpd.service
                  └─1579 proftpd: (accepting connections)

Apr 01 14:20:28 astechvm.local systemd[1]: Stopped ProFTPD F...
Apr 01 14:20:28 astechvm.local systemd[1]: Starting ProFTPD I...
Apr 01 14:20:28 astechvm.local systemd[1]: Can't open PID fil...
Apr 01 14:20:28 astechvm.local proftpd[1579]: 192.168.1.110 -
Apr 01 14:20:28 astechvm.local systemd[1]: Started ProFTPD F...
Hint: Some lines were ellipsized, use -l to show in full.
[root@astechvm ~]#
```

*systemctl enable proftpd*

Y verifica tu trabajo consultando el estado de ProFTPD:

*systemctl status proftpd*

Ahora, necesitamos modificar las políticas de SELinux, que tiene tres modos:

- Enforcing: SELinux permite el acceso basado en las reglas de política de SELinux.
- Permissive: SELinux solo registra acciones que se habrían denegado si se ejecutaran en modo de cumplimiento.
- Disabled: no se carga ninguna política de SELinux.

De manera predeterminada, en CentOS 7, SELinux está habilitado y en modo de aplicación. Se recomienda mantener SELinux en modo Enforcing, pero en algunos casos, es posible que deba configurarlo en un modo Permissive o deshabilitarlo por completo.

Para ver el estado actual de SELinux y la política de SELinux que se usa en su sistema, use el comando *sestatus*:

*sestatus*

Abrimos el archivo de configuración y configure el modo SELINUX en disabled/permissive/enforcing:

*vi /etc/selinux/config*

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Una vez completado, reiniciamos el sistema con *shutdown -r now*

Ahora, el servidor ProFTPD se ejecuta y escucha las conexiones, pero no está disponible para conexiones externas debido a la política de Firewall.

Para habilitar las conexiones externas, asegúrese de agregar una regla que abra el puerto 21, utilizando la utilidad del sistema *firewall-cmd*:

```
firewall-cmd --add-service=ftp
firewall-cmd --add-service=ftp --permanent
```

```
firewall-cmd --reload
```

Si no se ha creado usuarios de sistema en la instalación, podemos crearlos de esta manera:

```
useradd nombreUsuario
passwd contraseñaUsuario
```

Teniendo especial cuidado al elegir una contraseña robusta.

```
[root@astechvm ~]# useradd elisa
[root@astechvm ~]# passwd elisa
Changing password for user elisa.
New password:
BAD PASSWORD: The password fails the dictionary check -
ionary word
Retype new password:
passwd: all authentication tokens updated successfully.
```

Por ultimo, debemos permitir ProFTPD a través del cortafuegos.

Permitir el puerto FTP predeterminado, el puerto 21, a través de firewalld:

```
firewall-cmd --permanent --add-port=21/tcp
```

Permitir el puerto FTP predeterminado, puerto 21, a través de firewalld, permanente:

```
firewall-cmd --permanent --add-port=21/tcp
```

Y recargar el firewall:

```
firewall-cmd --reload
```

```
[root@astechvm ~]# firewall-cmd --add-port=21/tcp
Warning: ALREADY_ENABLED: '21:tcp' already in 'public'
success
[root@astechvm ~]# firewall-cmd --permanent --add-port=21/tcp
Warning: ALREADY_ENABLED: 21:tcp
success
[root@astechvm ~]# firewall-cmd --reload
success
```

Por último toca llevar a cabo la habilitación del modo FTP pasivo, que surge como resultado de problemas de conexión en el modo activo. El principal inconveniente del modo FTP activo es que si no hay una política específica en los cortafuegos del lado del cliente, es probable que se bloquee la conexión.

Cargamos el módulo del kernel: *modprobe ip\_conntrack\_ftp*

Agregamos el módulo para carga automática: *cat << EOF >*

```
/etc/modules-load.d/ip_conntrack_ftp.conf
```

```
ip_conntrack_ftp
```

```
EOF
```

---

```
[root@astechvm ~]# modprobe ip_conntrack_ftp
[root@astechvm ~]# cat << EOF > /etc/modules-load.d/ip_conntrack_ftp.
> ip_conntrack_ftp
> EOF
[root@astechvm ~]# systemctl restart systemd-modules-load.service
[root@astechvm ~]#
```

Reiniciamos el servicio del módulo: `systemctl restart systemd-modules-load.service`

Abrir puertos FTP en el cortafuegos:

```
iptables -I INPUT 6 -p tcp -m multiport --dports 20,21 -j ACCEPT
iptables -I INPUT 6 -p udp -m multiport --dports 20,21 -j ACCEPT
iptables -I INPUT 6 -p udp -m multiport --dports 20,21 -j ACCEPT
```

Comprobamos las reglas añadidas: `iptables -L INPUT -vn`

Habilitamos el inicio automático de ProFTPD en el arranque del sistema: `systemctl enable proftpd.service`

```
[root@astechvm ~]# iptables -I INPUT 6 -p tcp -m multiport --dports 20,21 -j ACCEPT
[root@astechvm ~]# iptables -I INPUT 6 -p udp -m multiport --dports 20,21 -j ACCEPT
[root@astechvm ~]# iptables -L INPUT -vn
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target  prot opt in  out  source        destination
on
 1918 175K ACCEPT   all  --  *    *      0.0.0.0/0      0.0.0.0/0
 0/0  0 ACCEPT   all  --  lo   *      0.0.0.0/0      0.0.0.0/0
 567 175K INPUT_direct all  --  *    *      0.0.0.0/0      0.0.0.0
 0/0  567 175K INPUT_ZONES_SOURCE all  --  *    *      0.0.0.0/0      0.0.0.0
 0/0  567 175K INPUT_ZONES all  --  *    *      0.0.0.0/0      0.0.0.0
 0  0 ACCEPT   udp  --  *    *      0.0.0.0/0      0.0.0.0/0
 0  0 ACCEPT   tcp  --  *    *      0.0.0.0/0      0.0.0.0/0
 0  0 ACCEPT   tcp  --  *    *      0.0.0.0/0      0.0.0.0/0
 0  0 ACCEPT   udp  --  *    *      0.0.0.0/0      0.0.0.0/0
 0  0 ACCEPT   tcp  --  *    *      0.0.0.0/0      0.0.0.0/0
 0  0 ACCEPT   multiport dports 20,21 -- *    *      0.0.0.0/0      0.0.0.0/0
 0  0 REJECT   all  --  *    *      0.0.0.0/0      0.0.0.0/0
reject-with icmp-host-prohibited
[root@astechvm ~]# systemctl enable proftpd.service
[root@astechvm ~]# systemctl start proftpd.service
```

Iniciamos el servidor ProFTPD: `systemctl start proftpd.service`

Con esto, ProFTP debería de funcionar correctamente.

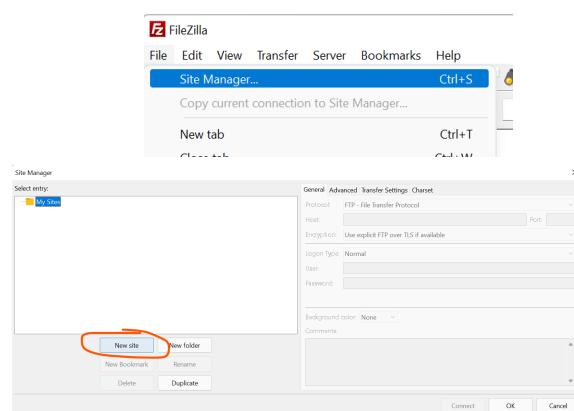
### Tarea 3.2 - Filezilla.

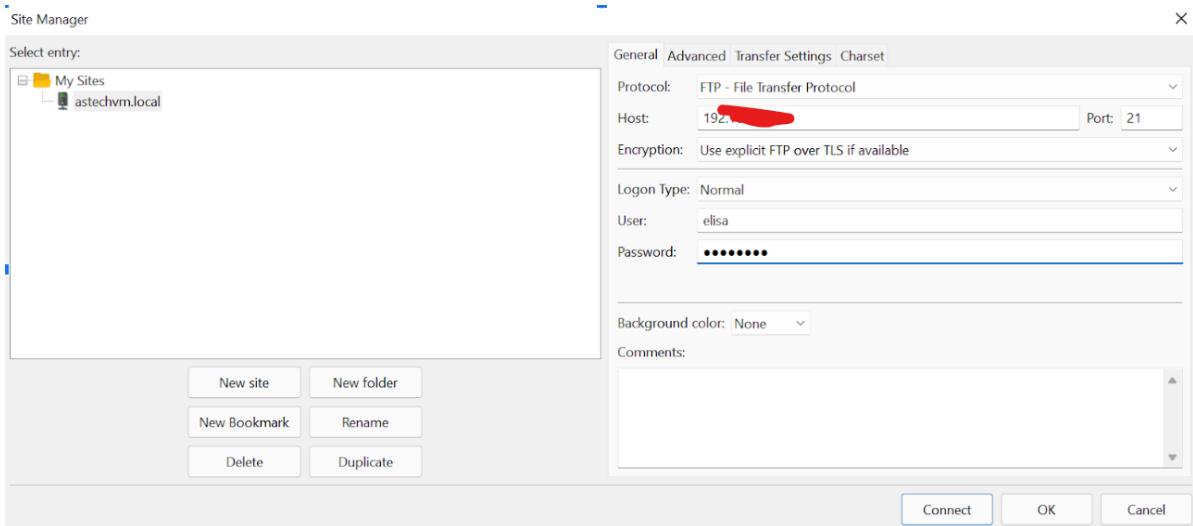
FileZilla Client es un cliente FTP, FTPS y SFTP multiplataforma rápido y confiable con muchas funciones útiles y una interfaz gráfica de usuario intuitiva.

Una vez descargado e instalado, procedemos a configurar nuestro sitio, que será el modo de acceso a nuestro directorio configurado para poder subir la aplicación Laravel.

Hacemos click en “File” > “Site Manager”.

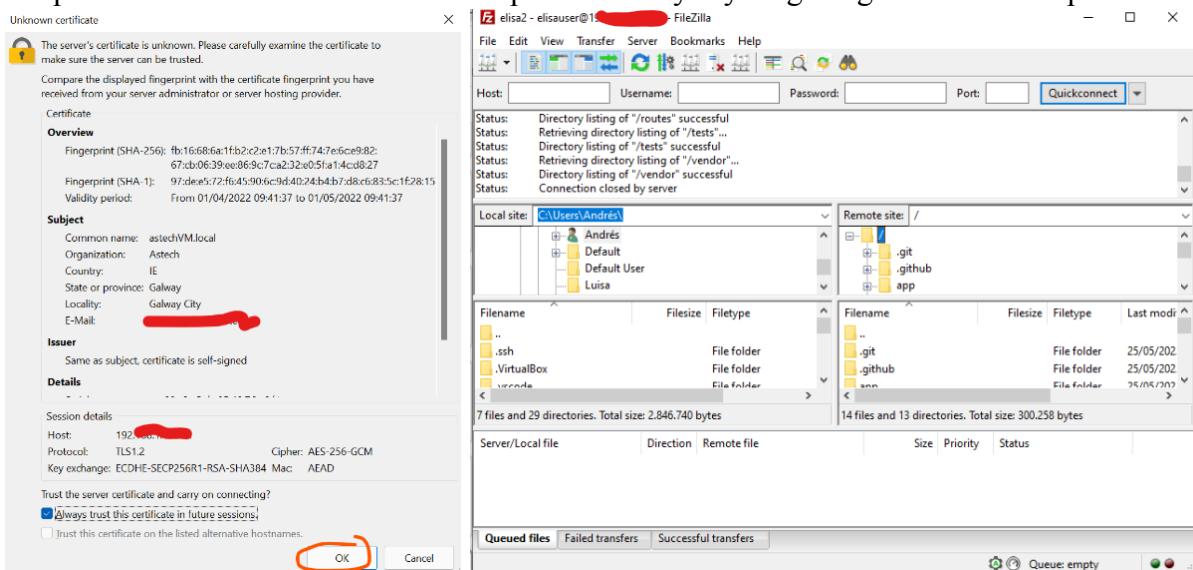
Luego, en “New Site”.





1. Configuramos el nombre del sitio.
2. Introducimos la IP pública del sitio.
3. Configuramos “logon” como Normal.
4. Introducimos usuario y contraseña.
5. Clic en “connect”.

Aceptamos el certificado marcando la opción “Always” y luego haga clic en “Accept”.



Y si sale todo bien, tendremos conexión con el servidor.

Ahora podemos navegar por el sistema. El lado izquierdo es nuestro sistema, el lado derecho es nuestro servidor. Podemos copiar, editar, arrastrar y soltar, entre muchas otras opciones.

Local site: C:\Users\andre\		Remote site: /var/www/html/laravel	
<span style="color: #ccc;">andre</span> <span style="color: #ccc;">Default</span> <span style="color: #ccc;">Default User</span> <span style="color: #ccc;">Public</span> <span style="color: #ccc;">webdriver</span> <span style="color: #ccc;">Windows</span>		<span style="color: #ccc;">/</span> <span style="color: #ccc;">bin</span> <span style="color: #ccc;">boot</span> <span style="color: #ccc;">dev</span> <span style="color: #ccc;">etc</span> <span style="color: #ccc;">ftpshare</span>	
Filename	Filesize	Filetype	Last modified
..			
.afirma		File folder	31/03/2022...
.config		File folder	05/03/2022...
.docker		File folder	16/03/2022...
.idlerc		File folder	21/03/2022...
.ssh		File folder	29/03/2022...
.VirtualBox		File folder	01/04/2022...
..			
.git		File fol...	30/03/20...
.github		File fol...	30/03/20...
app		File fol...	30/03/20...
bootstrap		File fol...	30/03/20...
config		File fol...	30/03/20...
database		File fol...	20/02/20...

Como vemos, ajustamos los parámetros anteriormente para que nuestro usuario se quede enjaulado en el directorio específico.

Para desconectarnos del servidor, debemos hacer clic aquí. →



The screenshot shows the FileZilla interface. The top menu bar includes File, Edit, View, Transfer, Server, Bookmarks, and Help. Below the menu is a toolbar with various icons for file operations like upload, download, and search. A status bar at the bottom displays the current connection status.

Host: [redacted] Username: [redacted] Disconnects from the current host

Status: Directory listing of "/var/www/html" successful

Status: Retrieving directory listing of "/var/www/html/laravel"...

Status: Directory listing of "/var/www/html/laravel" successful

Local site: C:\Users\andre\

Host: [redacted] Username: [redacted] Password: [redacted]

Status: Retrieving directory listing of "/var/www/html/laravel"...

Status: Directory listing of "/var/www/html/laravel" successful

Status: Disconnected from server

Ya estaría listo para subir nuestra aplicación Laravel al servidor.

Ahora toca trabajar en local, donde vamos a crear, en este proyecto, una API funcional que sea consumida por una aplicación de front-end.

## Tarea 4 - Desarrollo de una API en local.

En esta primera parte nos centraremos en la configuración, la API y la autenticación de Laravel.

Para crear el proyecto localmente vamos a utilizar Visual Studio Code, aunque se puede utilizar cualquier otro editor de código.

XAMPP es una herramienta de desarrollo que le permite probar su desarrollo web basado en PHP en su propia computadora sin necesidad de acceso a Internet. Si es un diseñador web o un desarrollador web que acaba de empezar, no necesita saber acerca de las configuraciones del servidor, XAMPP brinda una configuración completamente funcional desde el momento en que lo instala. Básicamente lo extraes y listo.

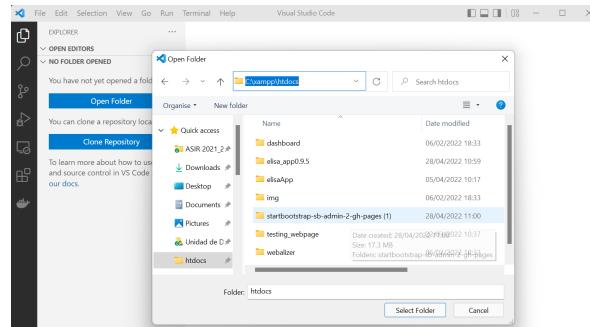
Composer es una herramienta para la gestión de dependencias en PHP. Permite declarar las bibliotecas de las que depende el proyecto y las administrará (instalará/actualizará) por el usuario.

### Tarea 4.1 - Preparación del proyecto.

En el código de Visual Studio

1. Abrimos la carpeta.
2. Seleccionamos la carpeta a utilizar. La ruta es  
`C:\xampp\htdocs`

Ahora es el momento de crear un proyecto Laravel.



*Control + n* para abrir la terminal.

Tenemos que instalar con Composer, los componentes de laravel:  
`composer global require laravel/installer`

```
PS C:\xampp\htdocs> composer global require laravel/installer
Changed current directory to C:/Users/andre/AppData/Roaming/Composer
Using version ^4.2 for laravel/installer
./composer.json has been updated
```

Con este comando creamos un nuevo proyecto:

*laravel new elisa*

PS C:\xampp\htdocs> **laravel new elisa2**

Elisa es el nombre de nuestro proyecto.

```
> @php artisan key:generate --ansi
Application key set successfully.

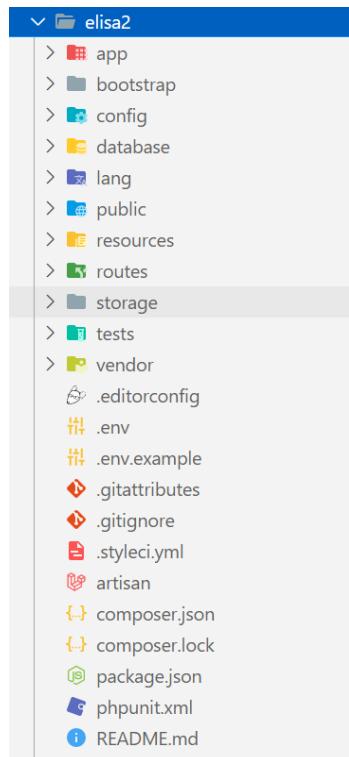
Application ready! Build something amazing.
PS C:\xampp\htdocs>
```

Una vez realizado el proceso, podemos ver que tenemos el proyecto listo para empezar a añadir nuestras modificaciones.

Al abrir la carpeta que se ha creado, podemos ver todo el contenido del nuevo proyecto.

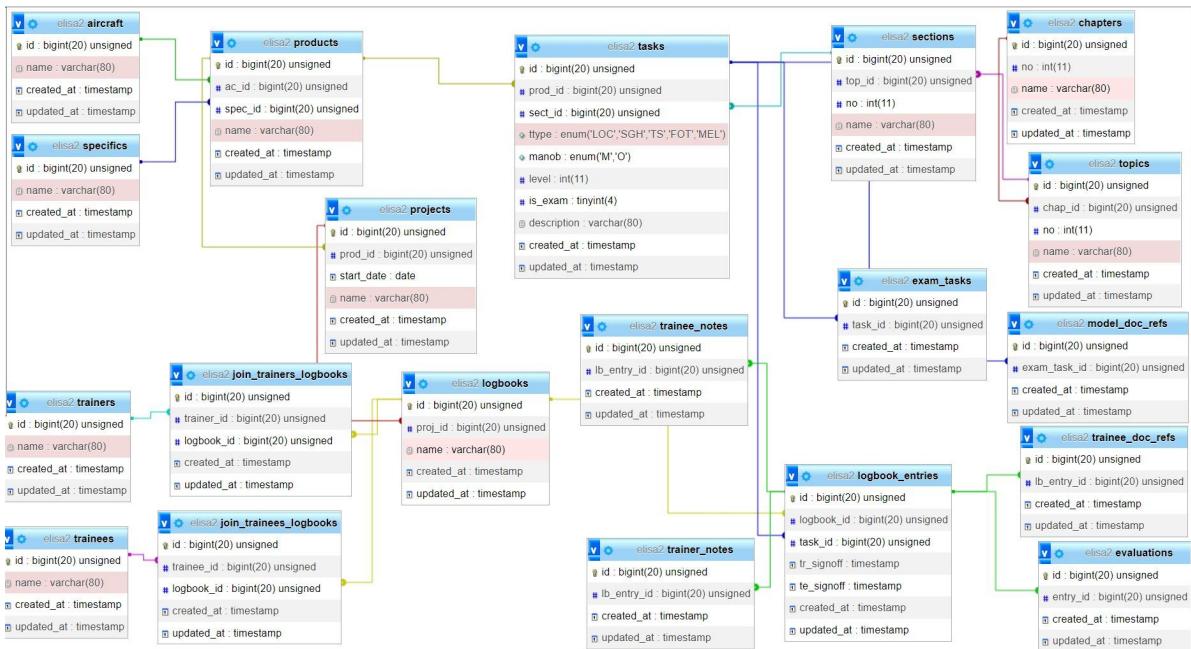
Para comprobar que carga nuestro proyecto Laravel, escribimos en el navegador:

<http://localhost/elisa/public/>



## Tarea 4.2 - Diseño de la bases de datos.

Con un programa de diagramas de flujo como LucidChart podremos diseñar un diagrama de flujo de lo que será nuestra base de datos. imprescindible para empezar a crear las migraciones.



## Tarea 4.3 - Gestión de las bases de datos.

Tenemos que especificarle a Laravel el trabajo con MySQL. Para ello, y con XAMPP iniciado, escribimos en el navegador:  
`localhost/phpmyadmin`

Y dentro de la interfaz, creamos una base de datos, llamada “elisa2”:

1. Haga clic en "Nuevo"
2. Poner el nombre.
3. Clic en “Crear”

Las migraciones le permiten modificar y compartir el esquema de la base de datos de la aplicación.

Cada modificación se guarda como registros (en caso de que sea necesario revertir los cambios).

Las podemos encontrar en “databases” > “migrations”.

Para crear una nueva migración tecleamos en la terminal:

```
php artisan make:migration create_tasks_table
```

donde `tasks` es el nombre de la tabla. cuando vamos a crear nuevas tablas, lo modificamos para que coincida con nuestro proyecto, por ejemplo: `create_tablelename_table`

De esta forma, seguimos las convenciones de Laravel.

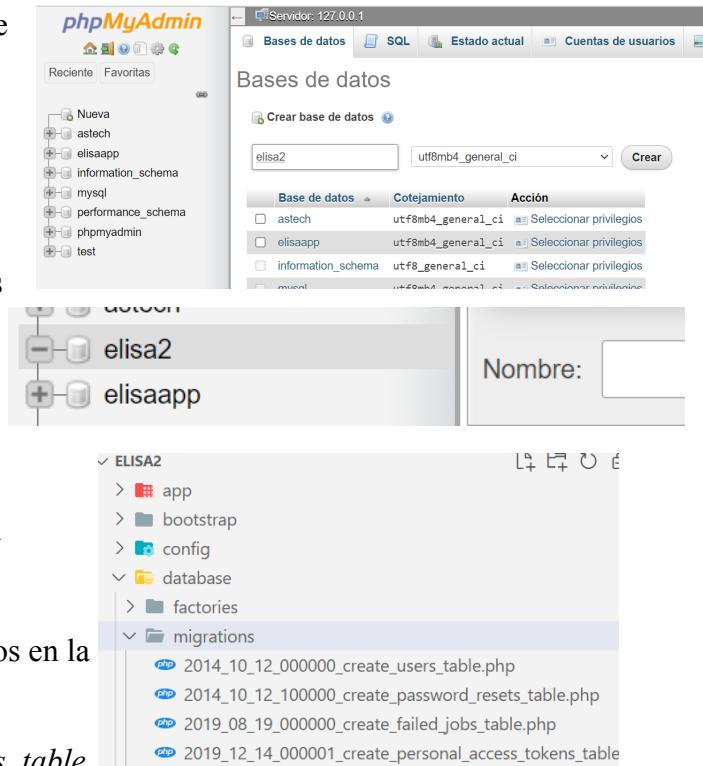
```
PS C:\xampp\htdocs\elisa2> php artisan make:migration create_tasks
Created Migration: 2022_05_06_122757_create_tasks

OPEN EDITORS
  × 2022_05_06_122757_create_tasks.php.d...
  · ELISA2
    > app
    > bootstrap
    > config
    < database
      > factories
      < migrations
        2014_10_12_000000_create_users_table.php
        2014_10_12_100000_create_password_resets_table.php
        2019_08_19_000000_create_failed_jobs_table.php
        2019_12_14_000001_create_personal_access_tokens_table.php
        2022_05_06_122757_create_tasks.php
      < seeders
        DatabaseSeeder.php
        .gitignore
    > lang
    > public
    > resources
    > routes
```

database > migrations > 2022\_05\_06\_122757\_create\_tasks.php > #Class#4c4e38cd > down

```

  9   /**
 10  * Run the migrations.
 11  *
 12  * @return void
 13  */
 14  public function up()
 15  {
 16      Schema::create('tasks', function (Blueprint $table) {
 17          $table->id();
 18          $table->timestamps();
 19      });
 20  }
 21
 22  /**
 23  * Reverse the migrations.
 24  *
 25  * @return void
 26  */
 27  public function down()
 28  {
 29      Schema::dropIfExists('tasks');
 30  }
```



The screenshot shows the phpMyAdmin interface with the following details:

- Servidor:** 127.0.0.1
- Bases de datos:** elisa2
- SQL:** utf8mb4\_general\_ci
- Estado actual:** Creado
- Cuentas de usuarios:** Creado
- Bases de datos:** elisa2, astech, elisaapp, information\_schema, mysql, performance\_schema, phpmyadmin, test
- Crear base de datos:** Nombre: elisa2, Cotejamiento: utf8mb4\_general\_ci, Acción: Crear
- Base de datos:** elisa2, Cotejamiento: utf8mb4\_general\_ci, Acción: Seleccionar privilegios
- elisa2:** Nombre: elisa2, Cotejamiento: utf8mb4\_general\_ci, Acción: Seleccionar privilegios
- elisaapp:** Nombre: elisaapp, Cotejamiento: utf8mb4\_general\_ci, Acción: Seleccionar privilegios
- elisa2:** Nombre: elisa2, Cotejamiento: utf8mb4\_general\_ci, Acción: Seleccionar privilegios
- elisaapp:** Nombre: elisaapp, Cotejamiento: utf8mb4\_general\_ci, Acción: Seleccionar privilegios
- ELISA2:** Nombre: ELISA2, Cotejamiento: utf8mb4\_general\_ci, Acción: Seleccionar privilegios
- Content:**
  - app
  - bootstrap
  - config
  - database
  - factories
  - migrations
    - 2014\_10\_12\_000000\_create\_users\_table.php
    - 2014\_10\_12\_100000\_create\_password\_resets\_table.php
    - 2019\_08\_19\_000000\_create\_failed\_jobs\_table.php
    - 2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table.php

Una vez creada la migración, tenemos que agregar los elementos que componen la tabla, en la función *up*:

Una vez hecho esto, ejecutamos el comando de migración *php artisan migrate*.

Después, *php artisan migrate:refresh* o *php artisan migrate:fresh* para recargar la base de datos con las modificaciones que hayamos hecho.

```

12  * @return void
13  */
14  public function up()
15  {
16      Schema::create('aircraft', function (Blueprint $table) {
17          $table->engine = "InnoDB";
18          $table->bigIncrements('id');
19          $table->string('name', 80)->nullable(false);
20      });
21  }
22  /**
23

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```

Migrated: 2014_10_12_000000_create_users_table (38.93ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (37.66ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (36.09ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (51.32ms)
Migrating: 2022_05_09_100834_create_aircraft
Migrated: 2022_05_09_100834_create_aircraft (19.10ms)
PS C:\xampp\htdocs\elisa2>

```

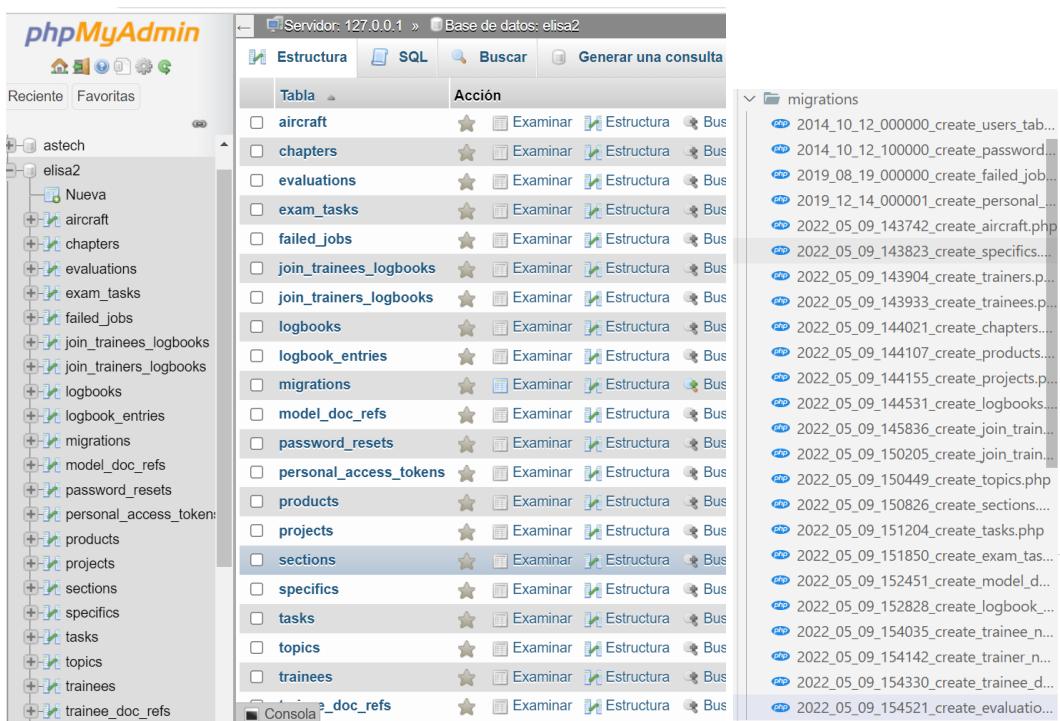
Luego, vamos a phpmyadmin y verificamos que efectivamente las migraciones se han realizado correctamente.



The screenshot shows the 'Estructura de tabla' (Table Structure) tab in phpMyAdmin. The table 'aircraft' has two columns:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
1	<b>id</b>	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
2	<b>name</b>	varchar(80)	utf8mb4_unicode_ci		No	Ninguna		

A continuación, muestro una captura de unas migraciones efectuadas correctamente:



The screenshot shows the phpMyAdmin interface connected to a local server (127.0.0.1) and a database named 'elisa2'. The left sidebar lists various tables under the schema 'elisa2'. The main area displays the 'migrations' table, which contains a list of migration scripts. Each entry includes a timestamp and a file name, such as '2014\_10\_12\_100000\_create\_password...' and '2022\_05\_09\_154521\_create\_evaluatio...'. The table has two columns: 'Tabla' (Table) and 'Acción' (Action).

Tabla	Acción
aircraft	Examinar Estructura Buscar
chapters	Examinar Estructura Buscar
evaluations	Examinar Estructura Buscar
exam_tasks	Examinar Estructura Buscar
failed_jobs	Examinar Estructura Buscar
join_trainees_logbooks	Examinar Estructura Buscar
join_trainers_logbooks	Examinar Estructura Buscar
logbooks	Examinar Estructura Buscar
logbook_entries	Examinar Estructura Buscar
migrations	Examinar Estructura Buscar
model_doc_refs	Examinar Estructura Buscar
password_resets	Examinar Estructura Buscar
personal_access_tokens	Examinar Estructura Buscar
products	Examinar Estructura Buscar
projects	Examinar Estructura Buscar
sections	Examinar Estructura Buscar
specifics	Examinar Estructura Buscar
tasks	Examinar Estructura Buscar
topics	Examinar Estructura Buscar
trainees	Examinar Estructura Buscar
_doc_refs	Examinar Estructura Buscar

**migrations**

- 2014\_10\_12\_000000\_create\_users\_table...
- 2014\_10\_12\_100000\_create\_password...
- 2019\_08\_19\_000000\_create\_failed\_job...
- 2019\_12\_14\_000001\_create\_personal...
- 2022\_05\_09\_143742\_create\_aircraft.php
- 2022\_05\_09\_143823\_create\_specifics...
- 2022\_05\_09\_143904\_create\_trainers.p...
- 2022\_05\_09\_143933\_create\_trainees.p...
- 2022\_05\_09\_144021\_create\_chapters...
- 2022\_05\_09\_144107\_create\_products...
- 2022\_05\_09\_144155\_create\_projects.p...
- 2022\_05\_09\_144531\_create\_logbooks...
- 2022\_05\_09\_145836\_create\_join\_train...
- 2022\_05\_09\_150205\_create\_join\_train...
- 2022\_05\_09\_150449\_create\_topics.php
- 2022\_05\_09\_150826\_create\_sections...
- 2022\_05\_09\_151204\_create\_tasks.php
- 2022\_05\_09\_151850\_create\_exam\_tas...
- 2022\_05\_09\_152451\_create\_model\_d...
- 2022\_05\_09\_152828\_create\_logbook...
- 2022\_05\_09\_154035\_create\_trainee\_n...
- 2022\_05\_09\_154142\_create\_trainer\_n...
- 2022\_05\_09\_154330\_create\_trainee\_d...
- 2022\_05\_09\_154521\_create\_evaluatio...

#### Tarea 4.4 - Inserción de datos de prueba para la base de datos.

A continuación, se adjuntan los distintos *inserts* para poder hacer pruebas con nuestra API más adelante.

Se ejecutarán en comando o mediante PHPMyAdmin, según conveniencia.

Aircraft:

```
INSERT INTO `aircraft` (`id`, `name`, `created_at`, `updated_at`) VALUES
(1, 'A330', '2022-06-07 15:39:55', '2022-06-07 15:39:55'),
(2, 'A380', '2022-06-07 15:39:55', '2022-06-07 15:39:55'),
(3, 'B737', '2022-06-07 15:39:55', '2022-06-07 15:39:55');
```

Chapters:

```
INSERT INTO `chapters` (`id`, `no`, `name`, `created_at`, `updated_at`) VALUES
(1, 21, 'chapter 21', '2022-06-07 15:41:44', '2022-06-07 15:41:44'),
```

(2, 24, 'chapter 24', '2022-06-07 15:41:44', '2022-06-07 15:41:44'),  
(3, 31, 'chapter 31', '2022-06-07 15:41:44', '2022-06-07 15:41:44'),  
(4, 18, 'Example', '2022-06-09 11:44:45', '2022-06-09 11:44:45'),  
(5, 90, 'Testing again', '2022-06-09 12:38:57', '2022-06-09 12:38:57');

#### Exam\_tasks:

```
INSERT INTO `exam_tasks` (`id`, `task_id`, `created_at`, `updated_at`) VALUES  
(2, 1, '2022-06-13 13:00:29', '2022-06-13 13:00:29'),(3, 2, '2022-06-13 13:00:35',  
'2022-06-13 13:00:35');
```

#### Join\_trainees\_logbooks

```
INSERT INTO `join_trainees_logbooks` (`id`, `trainee_id`, `logbook_id`, `created_at`,  
`updated_at`) VALUES  
(1, 1, 1, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(2, 2, 2, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(3, 3, 3, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(4, 4, 4,  
'2022-06-07 16:16:40', '2022-06-07 16:16:40'),  
(5, 5, 5, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(6, 6, 6, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(7, 7, 7, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(8, 8, 8,  
'2022-06-07 16:16:40', '2022-06-07 16:16:40'),(9, 9, 9, '2022-06-07 16:16:40', '2022-06-07  
16:16:40'),(10, 10, 10, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(11, 11, 11,  
'2022-06-07 16:16:40', '2022-06-07 16:16:40'),(12, 12, 12, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(13, 13, 13, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(14, 14,  
14, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(15, 15, 15, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(16, 16, 16, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(17, 17,  
17, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(18, 18, 18, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(19, 19, 19, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(20, 20,  
20, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(21, 21, 21, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(22, 22, 22, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(23, 23,  
23, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(24, 24, 24, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(25, 25, 25, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(26, 26,  
26, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(27, 27, 27, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(28, 28, 28, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(29, 29,  
29, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(30, 30, 30, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(31, 31, 31, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(32, 32, 3,  
'2022-06-07 16:16:40', '2022-06-07 16:16:40'),(33, 33, 33, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(34, 34, 34, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(35, 35,  
35, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(36, 36, 36, '2022-06-07 16:16:40',  
'2022-06-07 16:16:40'),(37, 10, 37, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(38, 11,  
38, '2022-06-07 16:16:40', '2022-06-07 16:16:40'),(39, 12, 39, '2022-06-07 16:16:40',
```

'2022-06-07 16:16:40'),(40, 1, 1, '2022-06-12 11:10:51', '2022-06-12 11:10:51'),(41, 3, 3, '2022-06-12 11:11:26', '2022-06-12 11:11:26');

join\_trainers\_logbooks:

```
INSERT INTO `join_trainers_logbooks` ('id', `trainer_id`, `logbook_id`, `created_at`, `updated_at`) VALUES
(1, 1, 1, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(2, 2, 1, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(3, 1, 2, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(4, 2, 2, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(5, 1, 3, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(6, 2, 3, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(7, 3, 4, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(8, 4, 4, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(9, 3, 5, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(10, 4, 5, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(11, 3, 6, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(12, 4, 6, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(13, 5, 7, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(14, 6, 7, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(15, 5, 8, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(16, 6, 8, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(17, 5, 9, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(18, 6, 9, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(19, 7, 10, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(20, 8, 10, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(21, 7, 11, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(22, 8, 11, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(23, 7, 12, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(24, 8, 12, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(25, 9, 13, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(26, 10, 13, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(27, 9, 14, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(28, 10, 14, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(29, 9, 15, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(30, 10, 15, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(31, 11, 16, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(32, 12, 16, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(33, 11, 17, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(34, 12, 17, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(35, 11, 18, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(36, 12, 18, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(37, 13, 19, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(38, 14, 19, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(39, 13, 20, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(40, 14, 20, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(41, 13, 21, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(42, 14, 21, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(43, 15, 22, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(44, 16, 22, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(45, 15, 23, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(46, 16, 23, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(47, 15, 24, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(48, 16, 24, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(49, 17, 25, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(50, 18, 25, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(51, 17, 26, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(52, 18, 26, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),
```

'2022-06-07 16:14:55'),(53, 17, 27, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(54, 18, 27, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(55, 19, 28, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(56, 20, 28, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(57, 19, 29, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(58, 20, 29, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(59, 19, 30, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(60, 20, 30, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(61, 21, 31, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(62, 22, 31, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(63, 21, 32, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(64, 22, 32, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(65, 21, 33, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(66, 22, 33, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(67, 23, 34, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(68, 24, 34, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(69, 23, 35, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(70, 24, 35, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(71, 23, 36, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(72, 24, 36, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(73, 3, 37, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(74, 4, 37, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(75, 3, 38, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(76, 4, 38, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(77, 3, 39, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(78, 4, 39, '2022-06-07 16:14:55', '2022-06-07 16:14:55'),(79, 3, 3, '2022-06-12 11:15:39', '2022-06-12 11:15:39');

### logbooks:

```
INSERT INTO `logbooks` (`id`, `proj_id`, `name`, `created_at`, `updated_at`) VALUES(1, 1, 'logbook 1 pr 1 tee 1 trs 1 2', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(2, 1, 'logbook 2 pr 1 tee 2 trs 1 2', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(3, 1, 'logbook 3 pr 1 tee 3 trs 1 2', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(4, 2, 'logbook 4 pr 2 tee 4 trs 3 4', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(5, 2, 'logbook 5 pr 2 tee 5 trs 3 4', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(6, 2, 'logbook 6 pr 2 tee 6 trs 3 4', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(7, 3, 'logbook 7 pr 3 tee 7 trs 5 6', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(8, 3, 'logbook 8 pr 3 tee 8 trs 5 6', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(9, 3, 'logbook 9 pr 3 tee 9 trs 5 6', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(10, 4, 'logbook 10 pr 4 tee 10 trs 7 8', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(11, 4, 'logbook 11 pr 4 tee 11 trs 7 8', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(12, 4, 'logbook 12 pr 4 tee 12 trs 7 8', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(13, 5, 'logbook 13 pr 5 tee 13 trs 9 10', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(14, 5, 'logbook 14 pr 5 tee 14 trs 9 10', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(15, 5, 'logbook 15 pr 5 tee 15 trs 9 10', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(16, 6, 'logbook 16 pr 6 tee 16 trs 11 12', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(17, 6, 'logbook 17 pr 6 tee 17 trs 11 12', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(18, 6, 'logbook 18 pr 6 tee 18 trs 11 12', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(19, 7, 'logbook 19 pr 7 tee 19 trs 13 14', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(20, 7, 'logbook 20 pr 7 tee 20 trs 13 14', '2022-06-07 16:11:13', '2022-06-07 16:11:13');
```

'2022-06-07 16:11:13', '2022-06-07 16:11:13'),(21, 7, 'logbook 21 pr 7 tee 21 trs 13 14', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(22, 8, 'logbook 22 pr 8 tee 22 trs 15 16', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(23, 8, 'logbook 23 pr 8 tee 23 trs 15 16', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(24, 8, 'logbook 24 pr 8 tee 24 trs 15 16', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(25, 9, 'logbook 25 pr 9 tee 25 trs 17 18', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(26, 9, 'logbook 26 pr 9 tee 26 trs 17 18', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(27, 9, 'logbook 27 pr 9 tee 27 trs 17 18', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(28, 10, 'logbook 28 pr 10 tee 28 trs 19 20', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(29, 10, 'logbook 29 pr 10 tee 29 trs 19 20', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(30, 10, 'logbook 30 pr 10 tee 30 trs 19 20', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(31, 11, 'logbook 31 pr 11 tee 31 trs 21 22', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(32, 11, 'logbook 32 pr 11 tee 32 trs 21 22', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(33, 11, 'logbook 33 pr 11 tee 33 trs 21 22', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(34, 12, 'logbook 34 pr 12 tee 34 trs 23 24', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(35, 12, 'logbook 35 pr 12 tee 35 trs 23 24', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(36, 12, 'logbook 36 pr 12 tee 36 trs 23 24', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(37, 9, 'logbook 37 pr 9 tee 10 trs 3 4', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(38, 9, 'logbook 38 pr 9 tee 11 trs 3 4', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(39, 9, 'logbook 39 pr 9 tee 12 trs 3 4', '2022-06-07 16:11:13', '2022-06-07 16:11:13'),(42, 10, 'logbook test', '2022-06-13 07:40:50', '2022-06-13 07:40:50');

#### logbook\_entries:

```
INSERT INTO `logbook_entries` (`id`, `logbook_id`, `task_id`, `tr_signoff`, `te_signoff`, `created_at`, `updated_at`) VALUES
(1, 5, 1, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(2, 5, 2, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(3, 5, 3, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(4, 5, 4, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(5, 5, 5, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(6, 5, 6, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(7, 5, 7, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
(8, 5, 8, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),
```

(9, 10, 9, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (10, 10, 10, '2022-02-15 08:55:42', '2022-02-15 07:50:10', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (11, 10, 11, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (12, 10, 12, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (13, 10, 13, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (14, 10, 14, '0000-00-00 00:00:00', '2022-02-15 09:11:23', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (15, 10, 15, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (16, 10, 16, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (17, 37, 33, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (18, 37, 34, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (19, 37, 35, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (20, 37, 36, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (21, 37, 37, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (22, 37, 38, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (23, 37, 39, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59'),  
 (24, 37, 40, '0000-00-00 00:00:00', '0000-00-00 00:00:00', '2022-06-07 16:50:59', '2022-06-07 16:50:59');

### Products:

```
INSERT INTO `products` (`id`, `ac_id`, `spec_id`, `name`, `created_at`, `updated_at`)
VALUES
(1, 1, 1, 'A330 RR-Trent-700 B1 Practical Training', '2022-06-07 16:06:52', '2022-06-07 16:06:52'),
(2, 1, 2, 'A330 PW-4000 B1 Practical Training', '2022-06-07 16:06:52', '2022-06-07 16:06:52'),
```

(3, 2, 3, 'A380 RR-Trent-900 B1 Practical Training', '2022-06-07 16:06:52', '2022-06-07 16:06:52'),  
(4, 2, 4, 'A380 GP-7200 B2 Practical Training', '2022-06-07 16:06:52', '2022-06-07 16:06:52'),  
(5, 3, 5, 'B737 CFM-56 B1 & B2 Practical Training', '2022-06-07 16:06:52', '2022-06-07 16:06:52'),  
(6, 3, 6, 'B737 CFM-LEAP-1B B2 Practical Training', '2022-06-07 16:06:52', '2022-06-07 16:06:52');

Projects:

```
INSERT INTO `projects` (`id`, `prod_id`, `start_date`, `name`, `created_at`, `updated_at`)
VALUES
(1, 1, '2021-09-01', 'A330 RR-Trent Practical Training Paris Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(2, 1, '2021-09-01', 'A330 RR-Trent Practical Training Madrid Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(3, 2, '2021-09-01', 'A330 PW-4000 Practical Training Paris Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(4, 2, '2021-09-01', 'A330 PW-4000 Practical Training Madrid Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(5, 3, '2021-09-01', 'A380 RR-Trent-900 B2 Practical Training Paris Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(6, 3, '2021-09-01', 'A380 RR-Trent-900 B2 Practical Training Madrid Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(7, 4, '2021-09-01', 'A380 GP-7200 B2 Practical Training Paris Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(8, 4, '2021-09-01', 'A380 GP-7200 B2 Practical Training Madrid Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(9, 5, '2021-09-01', 'B737 CFM-56 B1 & B2 Practical Training Frankfurt Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(10, 5, '2021-09-01', 'B737 CFM-56 B1 & B2 Practical Training London Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(11, 6, '2021-09-01', 'B737 CFM-LEAP-1B B1 & B2 Practical Training Frankfurt Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25'),
(12, 6, '2021-09-01', 'B737 CFM-LEAP-1B B1 & B2 Practical Training London Q3 2021', '2022-06-07 16:08:25', '2022-06-07 16:08:25');
```

### sections:

```
INSERT INTO `sections` (`id`, `top_id`, `no`, `name`, `created_at`, `updated_at`) VALUES
(1, 1, 0, 'sec 0 in topic 0 chap 21', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(2, 2, 0, 'sec 0 in topic 10 chap 21', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(3, 3, 0, 'sec 0 in topic 20 chap 21', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(4, 4, 0, 'sec 0 in topic 30 chap 21', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(5, 5, 0, 'sec 0 in topic 40 chap 21', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(6, 6, 0, 'sec 0 in topic 50 chap 21', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(7, 7, 0, 'sec 0 in topic 0 chap 24', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(8, 8, 0, 'sec 0 in topic 10 chap 24', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(9, 9, 0, 'sec 0 in topic 20 chap 24', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(10, 10, 0, 'sec 0 in topic 30 chap 24', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(11, 11, 0, 'sec 0 in topic 40 chap 24', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(12, 12, 0, 'sec 0 in topic 50 chap 41', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(13, 13, 0, 'sec 0 in topic 0 chap 31', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(14, 14, 0, 'sec 0 in topic 10 chap 31', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(15, 15, 0, 'sec 0 in topic 20 chap 31', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(16, 16, 0, 'sec 0 in topic 30 chap 31', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(17, 17, 0, 'sec 0 in topic 40 chap 31', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(18, 18, 0, 'sec 0 in topic 40 chap 31', '2022-06-07 16:20:03', '2022-06-07 16:20:03'),
(19, 8, 10, 'sec 10 in topic 10 chap 24', '2022-06-07 16:20:03', '2022-06-07 16:20:03');
```

### specifics:

```
INSERT INTO `specifics` (`id`, `name`, `created_at`, `updated_at`) VALUES
(1, 'RR Trent-700', '2022-06-07 15:59:47', '2022-06-07 15:59:47'),
(2, 'PW 4000', '2022-06-07 15:59:47', '2022-06-07 15:59:47'),
(3, 'RR Trent-900', '2022-06-07 15:59:47', '2022-06-07 15:59:47'),
(4, 'GP 7200', '2022-06-07 15:59:47', '2022-06-07 15:59:47'),
(5, 'CFM 56', '2022-06-07 15:59:47', '2022-06-07 15:59:47'),
(6, 'CFM LEAP 1B', '2022-06-07 15:59:47', '2022-06-07 15:59:47');
```

### tasks:

```
INSERT INTO `tasks` (`id`, `prod_id`, `sect_id`, `ttype`, `manob`, `level`, `is_exam`, `description`, `created_at`, `updated_at`) VALUES
(1, 1, 1, 'TS', 'O', 3, 0, 'sec 0 in topic 0 chap 21 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),
(2, 1, 2, 'TS', 'M', 3, 0, 'sec 0 in topic 20 chap 21 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),
(3, 1, 3, 'SGH', 'M', 3, 0, 'sec 0 in topic 30 chap 21 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),
(4, 1, 4, 'TS', 'O', 3, 0, 'sec 10 in topic 10 chap 24 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19');
```

'2022-06-07 16:23:19'),(5, 1, 5, 'TS', 'M', 3, 0, 'sec 0 in topic 20 chap 24 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(6, 1, 6, 'SGH', 'O', 3, 0, 'sec 0 in topic 40 chap 24 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(7, 1, 7, 'MEL', 'O', 3, 0, 'sec 0 in topic 30 chap 31 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(8, 1, 8, 'MEL', 'M', 3, 0, 'sec 0 in topic 40 chap 31 prod 1', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(9, 2, 1, 'LOC', 'M', 3, 0, 'sec 0 in topic 0 chap 21 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(10, 2, 2, 'TS', 'M', 3, 0, 'sec 0 in topic 20 chap 21 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(11, 2, 3, 'FOT', 'O', 3, 0, 'sec 0 in topic 30 chap 21 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(12, 2, 4, 'TS', 'M', 3, 0, 'sec 10 in topic 10 chap 24 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(13, 2, 5, 'MEL', 'O', 3, 0, 'sec 0 in topic 20 chap 24 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(14, 2, 6, 'SGH', 'M', 3, 0, 'sec 0 in topic 40 chap 24 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(15, 2, 7, 'LOC', 'M', 3, 0, 'sec 0 in topic 30 chap 31 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(16, 2, 8, 'LOC', 'O', 3, 0, 'sec 0 in topic 40 chap 31 prod 2', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(17, 3, 1, 'MEL', 'O', 3, 0, 'sec 0 in topic 0 chap 21 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(18, 3, 2, 'LOC', 'M', 3, 0, 'sec 0 in topic 20 chap 21 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(19, 3, 3, 'FOT', 'O', 3, 0, 'sec 0 in topic 30 chap 21 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(20, 3, 4, 'LOC', 'M', 3, 0, 'sec 10 in topic 10 chap 24 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(21, 3, 5, 'FOT', 'M', 3, 0, 'sec 0 in topic 20 chap 24 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(22, 3, 6, 'FOT', 'O', 3, 0, 'sec 0 in topic 40 chap 24 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(23, 3, 7, 'SGH', 'M', 3, 0, 'sec 0 in topic 30 chap 31 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(24, 3, 8, 'LOC', 'M', 3, 0, 'sec 0 in topic 40 chap 31 prod 3', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(25, 4, 1, 'MEL', 'M', 3, 0, 'sec 0 in topic 0 chap 21 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(26, 4, 2, 'TS', 'O', 3, 0, 'sec 0 in topic 20 chap 21 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(27, 4, 3, 'SGH', 'M', 3, 0, 'sec 0 in topic 30 chap 21 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(28, 4, 4, 'MEL', 'M', 3, 0, 'sec 10 in topic 10 chap 24 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(29, 4, 5, 'MEL', 'M', 3, 0, 'sec 0 in topic 20 chap 24 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(30, 4, 6, 'TS', 'O', 3, 0, 'sec 0 in topic 40 chap 24 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(31, 4, 7, 'TS', 'O', 3, 0, 'sec 0 in topic 30 chap 31 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(32, 4, 8, 'FOT', 'M', 3, 0, 'sec 0 in topic 40 chap 31 prod 4', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(33, 5, 1, 'SGH', 'O', 3, 0, 'sec 0 in topic 0 chap 21 prod 5', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(34, 5, 2, 'FOT', 'O', 3, 0, 'sec 0 in topic 20 chap 21 prod 5', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(35, 5, 3, 'SGH', 'O', 3, 0, 'sec 0 in topic 30 chap 21 prod 5', '2022-06-07 16:23:19'),(36, 5, 4, 'MEL', 'M', 3, 0, 'sec 10 in topic 10 chap 24 prod 5', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(37, 5, 5, 'LOC', 'O', 3, 0, 'sec 0 in topic 20 chap 24 prod 5', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(38, 5, 6, 'FOT', 'M', 3, 0, 'sec 0 in topic 40 chap 24 prod 5', '2022-06-07 16:23:19', '2022-06-07 16:23:19'),(39, 5, 7, 'LOC', 'O', 3, 0, 'sec 0 in topic 30 chap 31 prod 5', '2022-06-07 16:23:19', '2022-06-07

16:23:19'),(40, 5, 8, 'MEL', 'M', 3, 0, 'sec 0 in topic 40 chap 31 prod 5', '2022-06-07 16:23:19'),(41, 6, 1, 'LOC', 'M', 3, 0, 'sec 0 in topic 0 chap 21 prod 6', '2022-06-07 16:23:19'),(42, 6, 2, 'FOT', 'M', 3, 0, 'sec 0 in topic 20 chap 21 prod 6', '2022-06-07 16:23:19'),(43, 6, 3, 'TS', 'M', 3, 0, 'sec 0 in topic 30 chap 21 prod 6', '2022-06-07 16:23:19'),(44, 6, 4, 'FOT', 'M', 3, 0, 'sec 10 in topic 10 chap 24 prod 6', '2022-06-07 16:23:19'),(45, 6, 5, 'FOT', 'M', 3, 0, 'sec 0 in topic 20 chap 24 prod 6', '2022-06-07 16:23:19'),(46, 6, 6, 'SGH', 'M', 3, 0, 'sec 0 in topic 40 chap 24 prod 6', '2022-06-07 16:23:19'),(47, 6, 7, 'MEL', 'O', 3, 0, 'sec 0 in topic 30 chap 31 prod 6', '2022-06-07 16:23:19'),(48, 6, 8, 'TS', 'M', 3, 0, 'sec 0 in topic 40 chap 31 prod 6', '2022-06-07 16:23:19'),(49, 6, 9, 'FOT', 'M', 3, 0, 'sec 0 in topic 30 chap 31 prod 6', '2022-06-07 16:23:19');

#### topics:

```
INSERT INTO `topics` (`id`, `chap_id`, `no`, `name`, `created_at`, `updated_at`) VALUES
(1, 1, 0, 'topic 0 in chap 21', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(2, 1, 10, 'topic 10 in chap 21', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(3, 1, 20, 'topic 20 in chap 21', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(4, 1, 30, 'topic 30 in chap 21', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(5, 1, 40, 'topic 40 in chap 21', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(6, 1, 50, 'topic 50 in chap 21', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(7, 2, 0, 'topic 0 in chap 24', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(8, 2, 10, 'topic 10 in chap 24', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(9, 2, 20, 'topic 20 in chap 24', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(10, 2, 30, 'topic 30 in chap 24i', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(11, 2, 40, 'topic 40 in chap 24', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(12, 2, 50, 'topic 50 in chap 24', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(13, 3, 0, 'topic 0 in chap 31', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(14, 3, 10, 'topic 10 in chap 31', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(15, 3, 20, 'topic 20 in chap 31', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(16, 3, 30, 'topic 30 in chap 31', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(17, 3, 40, 'topic 40 in chap 31', '2022-06-07 16:18:43', '2022-06-07 16:18:43'),
(18, 3, 50, 'topic 50 in chap 31', '2022-06-07 16:18:43', '2022-06-07 16:18:43');
```

#### trainees:

```
INSERT INTO `trainees` (`id`, `name`, `created_at`, `updated_at`) VALUES
(1, 'trainee-1', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(2, 'trainee-2', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(3, 'trainee-3', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(4, 'trainee-4', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(5, 'trainee-5', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(6, 'trainee-6', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(7, 'trainee-7', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),
(8,
```

'trainee-8', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(9, 'trainee-9', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(10, 'trainee-10', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(11, 'trainee-11', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(12, 'trainee-12', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(13, 'trainee-13', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(14, 'trainee-14', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(15, 'trainee-15', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(16, 'trainee-16', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(17, 'trainee-17', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(18, 'trainee-18', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(19, 'trainee-19', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(20, 'trainee-20', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(21, 'trainee-21', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(22, 'trainee-22', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(23, 'trainee-23', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(24, 'trainee-24', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(25, 'trainee-25', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(26, 'trainee-26', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(27, 'trainee-27', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(28, 'trainee-28', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(29, 'trainee-29', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(30, 'trainee-30', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(31, 'trainee-31', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(32, 'trainee-32', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(33, 'trainee-33', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(34, 'trainee-34', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(35, 'trainee-35', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(36, 'trainee-36', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(37, 'trainee-37', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(38, 'trainee-38', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(39, 'trainee-39', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(40, 'trainee-40', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(41, 'trainee-41', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(42, 'trainee-42', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(43, 'trainee-43', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(44, 'trainee-44', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(45, 'trainee-45', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(46, 'trainee-46', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(47, 'trainee-47', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(48, 'trainee-48', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(49, 'trainee-49', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(50, 'trainee-50', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(51, 'trainee-51', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(52, 'trainee-52', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(53, 'trainee-53', '2022-06-07 16:05:19', '2022-06-07 16:05:19'),(54, 'trainee-54', '2022-06-07 16:05:19', '2022-06-07 16:05:19');

trainee\_notes:

```
INSERT INTO `trainee_notes` (`id`, `lb_entry_id`, `created_at`, `updated_at`) VALUES
(3, 10, '2022-06-07 16:52:29', '2022-06-07 16:52:29'),
(4, 12, '2022-06-07 16:52:29', '2022-06-07 16:52:29');
```

### Trainers:

```
INSERT INTO `trainers` (`id`, `name`, `created_at`, `updated_at`) VALUES(1, 'trainer-1', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(2, 'trainer-2', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(3, 'trainer-3', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(4, 'trainer-4', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(5, 'trainer-5', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(6, 'trainer-6', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(7, 'trainer-7', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(8, 'trainer-8', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(9, 'trainer-9', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(10, 'trainer-10', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(11, 'trainer-11', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(12, 'trainer-12', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(13, 'trainer-13', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(14, 'trainer-14', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(15, 'trainer-15', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(16, 'trainer-16', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(17, 'trainer-17', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(18, 'trainer-18', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(19, 'trainer-19', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(20, 'trainer-20', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(21, 'trainer-21', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(22, 'trainer-22', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(23, 'trainer-23', '2022-06-07 16:02:51', '2022-06-07 16:02:51'),(24, 'trainer-24', '2022-06-07 16:02:51', '2022-06-07 16:02:51');
```

### **Tarea 4.5 - Creación de la API e introducción a POSTMAN.**

API es el acrónimo de Interfaz de programación de aplicaciones, que es un intermediario de software que permite que dos aplicaciones se comuniquen entre sí.

Una API web o API de servicio web es un conjunto de herramientas que permite a los desarrolladores enviar y recibir instrucciones y datos entre un servidor web y un navegador web, generalmente en formato JSON, para crear aplicaciones.

HTTP, un acrónimo de HyperText Transfer Protocol, es un protocolo que permite obtener recursos, como documentos HTML y datos JSON, a través de una conexión a Internet. Un protocolo se define como un sistema de reglas que define cómo se intercambian los datos dentro o entre computadoras. HTTP define un conjunto de métodos de solicitud que realizan diferentes acciones cuando se les llama. Los métodos de solicitud HTTP más utilizados son GET y POST.

Los métodos básicos de HTTP:

- GET permite recuperar datos a través de una URL (Uniform Resource Locator o Localizador de Recursos Uniforme)

- POST permite enviar información adicional con una solicitud a través de una URL para realizar ciertas acciones

Otros métodos HTTP son PUT, DELETE, HEAD, CONNECT, OPTIONS, TRACE y PATCH.

Para poder crear la API primero debemos crear los modelos, los cuales son un elemento que nos permite interactuar con nuestra base de datos de esta forma podemos realizar cualquier tipo de consulta a nuestra base de datos, por ejemplo: insertar, eliminar, modificar, mostrar .

*php artisan make:model Aircraft*

```
8   class Aircraft extends Model
9   {
10      use HasFactory;
11      protected $fillable = ['name'];
12 }
```

En el modelo lo llenamos con:

*protected \$fillable = ['name'];*

Los controladores son un mecanismo que nos permite agrupar la lógica de las solicitudes HTTP relacionadas y así organizar mejor nuestro código.

*php artisan make:controller Api\AircraftController --model=Aircraft*

Crea de forma automática todas las operaciones que se pueden hacer con la API.

Una vez realizado, lo primero que debemos hacer es declarar el controlador en la API, que se encuentra en “routes” > “api.php”.

*Route::apiResource('aircraft', AircraftController::class);*

Y en la parte superior, hacemos referencia al controlador de “aircraft”, en este caso.

*use App\Http\Controllers\Api\AircraftController;*

es >  api.php > ...

<?php

```
use App\Http\Controllers\Api\AircraftController;
use App\Http\Controllers\Api\SpecificsController;
use App\Http\Controllers\Api\TrainersController;
```

Una vez realizado, guardamos. Podemos comprobar con `php artisan route:list` que nos ha generado los métodos a usar, que será probados más tarde con Postman.

|                  |                                      |                                    |   |
|------------------|--------------------------------------|------------------------------------|---|
| <b>POST</b>      | <code>ignition/update-config</code>  | <code>ignition.updateConfig</code> | > Spatie\LaravelIgnition > UpdateConfigControl... |
| <b>GET HEAD</b>  | <code>api/aircraft</code>            | <code>aircraft.index</code>        | > Api\AircraftController@index                    |
| <b>POST</b>      | <code>api/aircraft</code>            | <code>aircraft.store</code>        | > Api\AircraftController@store                    |
| <b>GET HEAD</b>  | <code>api/aircraft/{aircraft}</code> | <code>aircraft.show</code>         | > Api\AircraftController@show                     |
| <b>PUT PATCH</b> | <code>api/aircraft/{aircraft}</code> | <code>aircraft.update</code>       | > Api\AircraftController@update                   |
| <b>DELETE</b>    | <code>api/aircraft/{aircraft}</code> | <code>aircraft.destroy</code>      | > Api\AircraftController@destroy                  |
| <b>GET HEAD</b>  | <code>api/user</code>                |                                    |   |
| <b>GET HEAD</b>  | <code>sanctum/csrf-cookie</code>     |                                    | Laravel\Sanctum > CsrfCookieController@show       |

Postman es una plataforma API para construir y usar APIs. Postman simplifica cada paso del ciclo de vida de la API y agiliza la colaboración para que se pueda crear mejores API, más rápida y eficientemente.

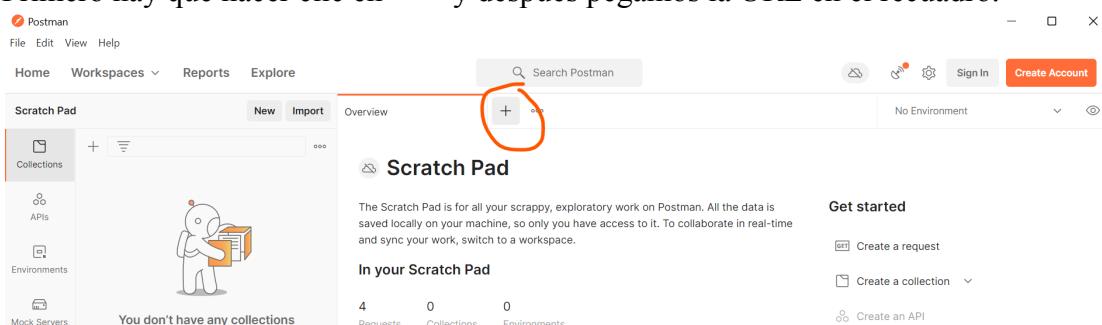
Postman se utiliza para múltiples tareas, entre las que destacaremos en esta ocasión las siguientes:

- Prueba de colecciones o catálogos de API tanto para front-end como para back-end.
- Organización de los servicios web en carpetas, funcionalidades y módulos.
- Permite gestionar el ciclo de vida (conceptualización y definición, desarrollo, seguimiento y mantenimiento) de nuestra API.
- Genera documentación de nuestras APIs.
- Trabaja con entornos (calidad, desarrollo, producción) y de esta forma es posible compartir información con el resto del equipo involucrado en el desarrollo a través de un entorno en la nube.

Ahora es el momento de probar la API con Postman. vamos a colocar la url en el programa.

`http://localhost/elisa2/public/api/aircraft`

Primero hay que hacer clic en “+” y despues pegamos la URL en el recuadro.

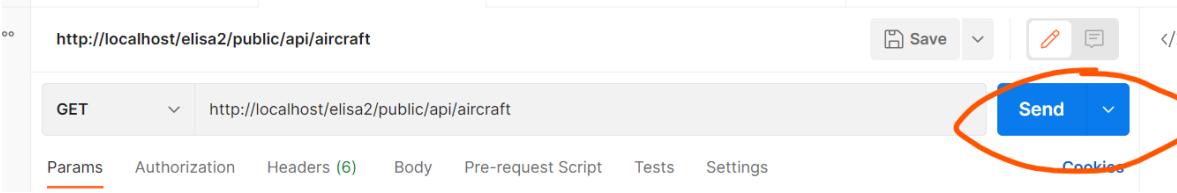


Luego en nuestro controlador, en el método “index()”, colocamos un “return” con la siguiente información, el cual devolverá un json para poder comprobar que realmente funciona.

```
return response()->json([
    'name' => []
]);
```

```
/*
 * @return \Illuminate\Http\Response
 */
public function index()
{
    return response()->json([
        'name' => []
    ]);
}
```

En el método, elegimos GET, que hará la petición al servidor y nos devolverá lo que le hemos designado.



The screenshot shows the Postman interface with a successful GET request to `http://localhost/elisa2/public/api/aircraft`. The 'Send' button is highlighted with a red circle. The response body is shown as:

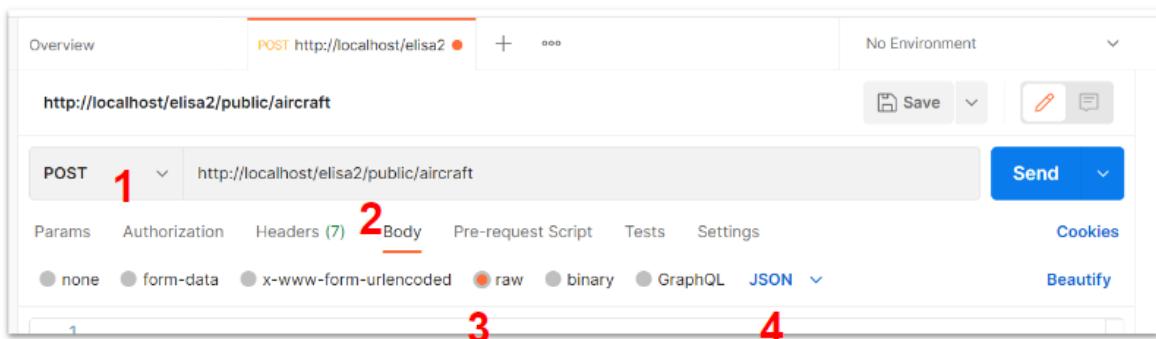
```
1 "name": []
```

La API funciona correctamente.

Ahora, es el turno de crear un registro. Comprobamos nuestras rutas y vemos a dónde tenemos que enviar la petición. Crear registros se vale del método POST.

|                    |   |
|--------------------|---|
| <b>GET   HEAD</b>  | <u>_ignition/health-check</u> .....     |
| <b>POST</b>        | <u>_ignition/update-config</u> ..... if |
| <b>GET   HEAD</b>  | <u>api/aircraft</u> .....               |
| <b>POST</b>        | <u>api/aircraft</u> .....               |
| <b>GET   HEAD</b>  | <u>api/aircraft/{aircraft}</u> .....    |
| <b>PUT   PATCH</b> | <u>api/aircraft/{aircraft}</u> .....    |
| <b>DELETE</b>      | <u>api/aircraft/{aircraft}</u> .....    |

Y lo colocamos dentro del programa en una nueva prueba.



1. Colocamos el método POST.
2. Haga clic en Body.
3. Haga clic en RAW.
4. Seleccione JSON.

En el campo Body, ponemos el siguiente código para generar un nuevo registro. Debe coincidir con los campos requeridos en nuestra tabla.

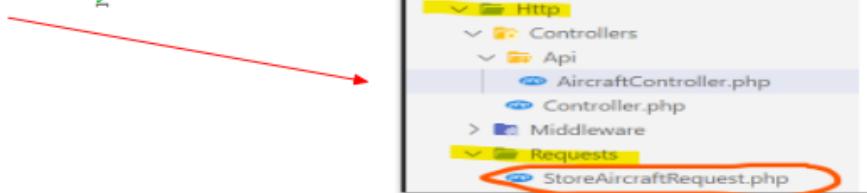
```
1 {  
2   "name": "testing API"  
3 }
```

Antes de enviar la petición, debemos crear una request, que es fichero que valida los campos requeridos y le aplica diversas reglas, como por ejemplo, la cantidad de caracteres, o si es requerido o no el campo a llenar. Es útil cuando queremos definir reglas y normas requeridas en los formularios. Si no se cumplen, no dejarán enviar la petición. Lo creamos con el comando:

`php artisan make:request StoreAircraftRequest`

Donde vemos que seguimos una convención, en este caso StoreNombreTablaRequest.

```
PS C:\xampp\htdocs\elisa2> php artisan make:request StoreAircraftRequest
Request created successfully.
```



Añadimos al Request las reglas de autorización, en el caso de cumplirse (valor authorize en true):

Como observamos, el campo “name” es obligatorio, con un máximo de 80 caracteres.

```

public function authorize()
{
    return true;
}

/**
 * Get the validation rules that apply to the request.
 *
 * @return array<string, mixed>
 */
public function rules()
{
    return [
        "name" => "required|max:80"
];
}

```

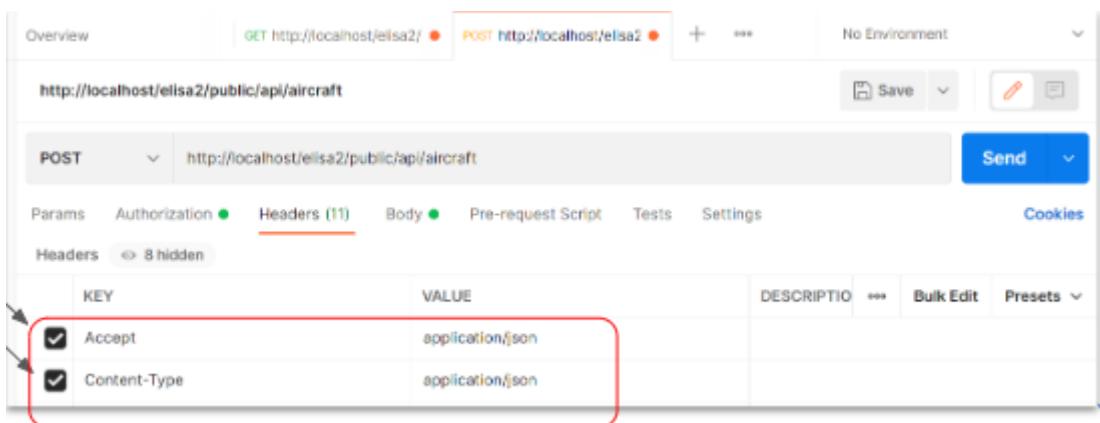
Vamos de nuevo al controlador, donde colocaremos en el método “store()” lo siguiente:

```

public function store(StoreAircraftRequest $request)
{
    dd($request->all());
}

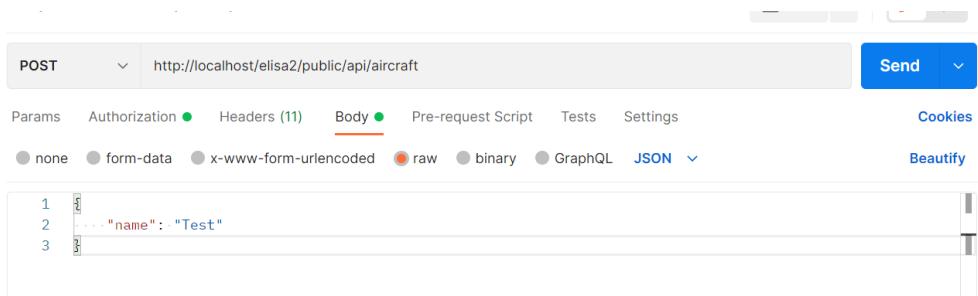
```

Y en Postman, seleccionamos los encabezados como se muestra a continuación:



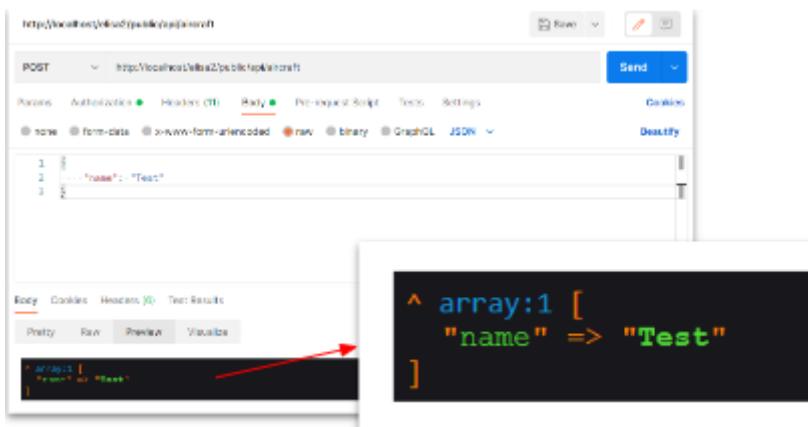
The screenshot shows the Postman interface for a POST request to `http://localhost/elisa2/public/api/aircraft`. The 'Headers' tab is selected, showing two entries: 'Accept' with value `application/json` and 'Content-Type' with value `application/json`. Both headers have a checked checkbox next to them. A red box highlights these two entries.

Dentro del cuerpo agregamos el campo para probar si funciona. Una vez hecho damos clic en Enviar.



The screenshot shows the Postman interface for the same POST request. The 'Body' tab is selected, showing a JSON object with a single field: `"name": "Test"`.

Una vez hecho. Hacemos clic en vista previa, y vemos que funciona.



Ahora es el momento de almacenar los datos. Modificamos el controlador.

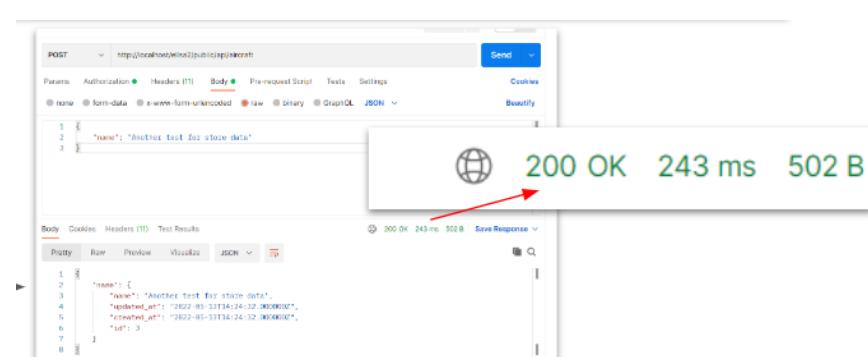
Método Store():

```
public function store(StoreAircraftRequest $request)
{
    $aircraft = Aircraft::create($request->all());

    return response()->json([
        'name' => $aircraft
    ], 200);
}
```

En Postman modificamos el cuerpo y enviamos de nuevo.

Después de enviar la solicitud, tenemos esto.



Ahora le toca el turno al método update ():

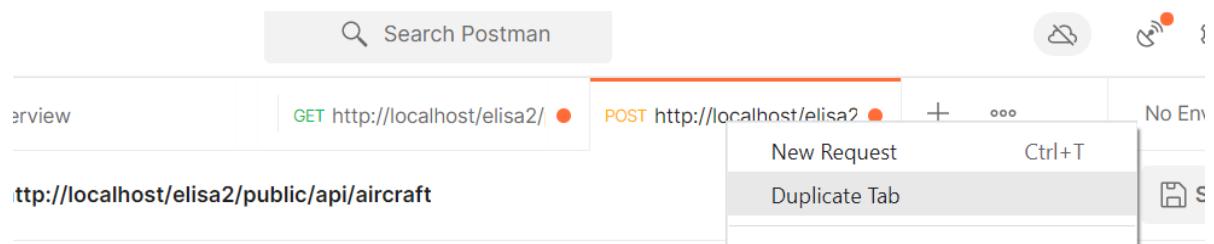
```
public function update(StoreAircraftRequest $request, Aircraft $aircraft)
{
    $aircraft->update($request->all());

    return response()->json([
        'name' => $aircraft
    ]);
}
```

Y vemos como funciona la ruta:

|           |                               |   |
|-----------|-------------------------------|---|
| GET HEAD  | api/aircraft/{aircraft} ..... | aircraft.show > Api\AircraftController@show       |
| PUT PATCH | api/aircraft/{aircraft} ..... | aircraft.update > Api\AircraftController@update   |
| DELETE    | api/aircraft/{aircraft} ..... | aircraft.destroy > Api\AircraftController@destroy |
| GET HEAD  | api/user .....                | Laravel\Sanctum > CsrfCookieController@show       |
| GET HEAD  | sanctum/csrf-cookie .....     |   |
| GET HEAD  | token .....                   |   |

Con el método PUT, necesitamos pasarle un parámetro, que por defecto es el ID. Una vez aclarado, vamos a Postman y duplicamos nuestra pestaña anterior:



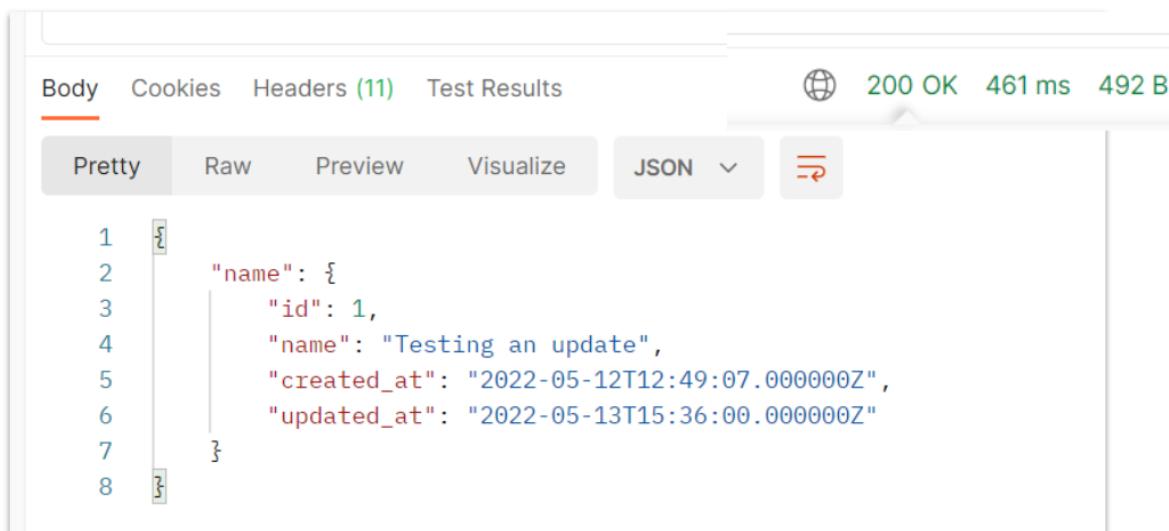
The screenshot shows the Postman interface with a list of requests in the background. In the foreground, a new request tab is being created by right-clicking on an existing tab labeled "http://localhost/elisa2/public/api/aircraft". A context menu is open with options: "New Request" (highlighted), "Duplicate Tab", and "Ctrl+T".

1. Ponemos el id correspondiente a los datos que vamos a actualizar.
2. Configuramos el método PUT
3. Realizamos una modificación del body.
4. Clic en enviar.



The screenshot shows the Postman request editor for a PUT request to <http://localhost/elisa2/public/api/aircraft/1>. The request method is set to PUT (highlighted with a red number 2). The URL is <http://localhost/elisa2/public/api/aircraft/1> (highlighted with a red number 1). The body contains the JSON payload: {"name": "Testing an update"} (highlighted with a red number 3). The "Send" button is highlighted with a red number 4.

El resultado es el siguiente:



Body Cookies Headers (11) Test Results

200 OK 461 ms 492 B

Pretty Raw Preview Visualize JSON ↻

```
1  {
2      "name": {
3          "id": 1,
4          "name": "Testing an update",
5          "created_at": "2022-05-12T12:49:07.000000Z",
6          "updated_at": "2022-05-13T15:36:00.000000Z"
7      }
8  }
```

Modificamos el método delete():

```
public function destroy(Aircraft $aircraft)
{
    $aircraft->delete();

    return response()->json([
        'name' => $aircraft
    ], 200);
}
```

Vemos las rutas.

| POST          |                                   |
|---------------|-----------------------------------|
| GET HEAD      | api/aircraft .....                |
| POST          | api/aircraft .....                |
| GET HEAD      | api/aircraft/{aircraft} ..        |
| PUT PATCH     | api/aircraft/{aircraft} ..        |
| <b>DELETE</b> | <b>api/aircraft/{aircraft} ..</b> |
| GET HEAD      | api/charts .....                  |

y en Postman, duplicamos la pestaña anterior. Seleccionamos el método DELETE y enviamos la petición.

http://localhost/elisa2/public/api/aircraft/1

Save Send

DELETE http://localhost/elisa2/public/api/aircraft/1

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1

Body Cookies Headers (11) Test Results 200 OK 443 ms 495 B Save Response

Podemos comprobar enviando de nuevo la petición o consultando en la bases de datos el elemento con el id que corresponda. En este caso, el elemento cuyo id es 1 fue correctamente eliminado.

Browse Structure SQL Search Insert Export Import

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

SELECT \* FROM `aircraft`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key

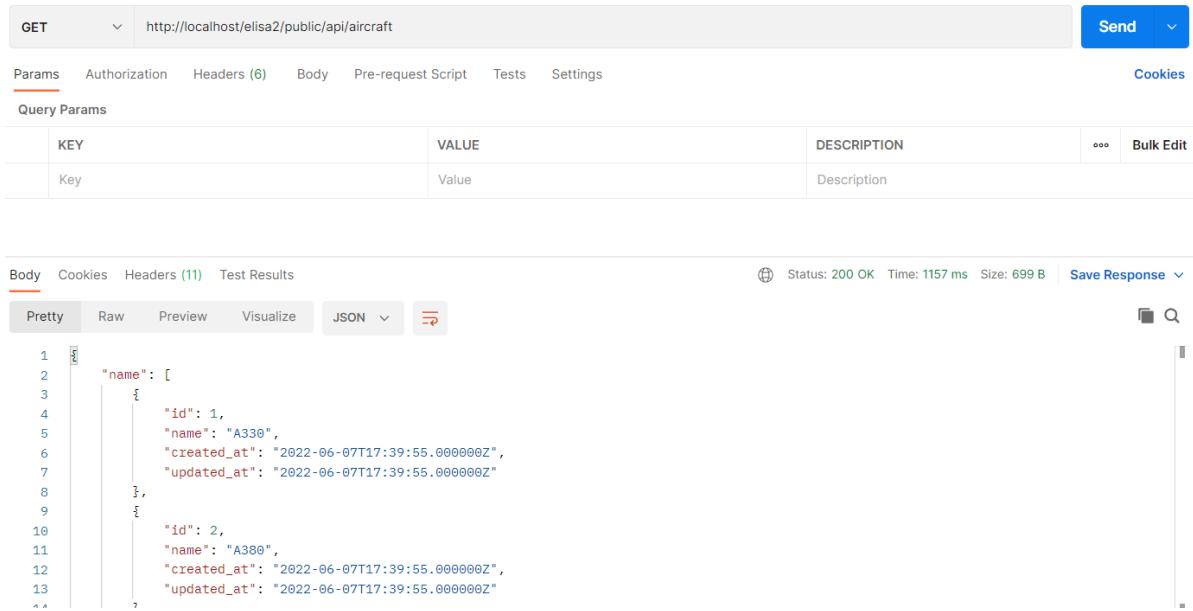
Extra options

| ← →   | id | name               | created_at          | updated_at          |
|---|----|--------------------|---------------------|---------------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2  | Testing API        | 2022-05-27 10:42:02 | 2022-05-27 10:42:02 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 3  | Testing API, again | 2022-05-27 10:43:33 | 2022-05-27 10:43:33 |

Por último, arreglaremos el primer método que modificamos al principio, el index():

```
public function index()
{
    $aircraft = Aircraft::all();
    return response()->json([
        'name' => $aircraft
    ]);
}
```

Y volvemos a ejecutar en una pestaña nueva con el método GET:



The screenshot shows a Postman request for `GET http://localhost/elisa2/public/api/aircraft`. The response body is a JSON array containing two aircraft objects:

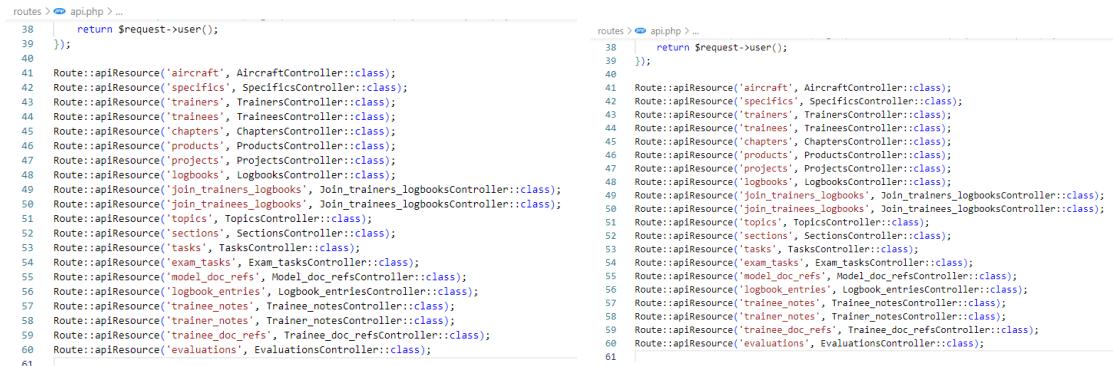
```

1
2   "name": [
3     {
4       "id": 1,
5       "name": "A330",
6       "created_at": "2022-06-07T17:39:55.000000Z",
7       "updated_at": "2022-06-07T17:39:55.000000Z"
8     },
9     {
10       "id": 2,
11       "name": "A380",
12       "created_at": "2022-06-07T17:39:55.000000Z",
13       "updated_at": "2022-06-07T17:39:55.000000Z"
14     }
  ]

```

Por lo tanto, podemos a partir de aquí diseñar las rutas correspondientes a nuestra aplicación.

Vamos a revisar como quedará nuestra API inicialmente:



```

routes > ➜ api.php > ...
38   return $request->user();
39 });
40
41 Route::apiResource('aircraft', AircraftController::class);
42 Route::apiResource('specifics', SpecificsController::class);
43 Route::apiResource('trainers', TrainersController::class);
44 Route::apiResource('trainees', TraineesController::class);
45 Route::apiResource('chapters', ChaptersController::class);
46 Route::apiResource('products', ProductsController::class);
47 Route::apiResource('projects', ProjectsController::class);
48 Route::apiResource('logbooks', LogbooksController::class);
49 Route::apiResource('join_trainers_logbooks', Join_trainers_logbooksController::class);
50 Route::apiResource('join_trainees_logbooks', Join_trainees_logbooksController::class);
51 Route::apiResource('topics', TopicsController::class);
52 Route::apiResource('sections', SectionsController::class);
53 Route::apiResource('tasks', TasksController::class);
54 Route::apiResource('exam_tasks', Exam_tasksController::class);
55 Route::apiResource('model_doc_refs', Model_doc_refsController::class);
56 Route::apiResource('logbook_entries', Logbook_entriesController::class);
57 Route::apiResource('trainee_notes', Trainee_notesController::class);
58 Route::apiResource('trainer_notes', Trainer_notesController::class);
59 Route::apiResource('trainee_doc_refs', Trainee_doc_refsController::class);
60 Route::apiResource('evaluations', EvaluationsController::class);
61

routes > ➜ api.php > ...
38   return $request->user();
39 });
40
41 Route::apiResource('aircraft', AircraftController::class);
42 Route::apiResource('specifics', SpecificsController::class);
43 Route::apiResource('trainers', TraineesController::class);
44 Route::apiResource('trainees', TraineesController::class);
45 Route::apiResource('chapters', ChaptersController::class);
46 Route::apiResource('products', ProductsController::class);
47 Route::apiResource('projects', ProjectsController::class);
48 Route::apiResource('logbooks', LogbooksController::class);
49 Route::apiResource('join_trainers_logbooks', Join_trainers_logbooksController::class);
50 Route::apiResource('join_trainees_logbooks', Join_trainees_logbooksController::class);
51 Route::apiResource('topics', TopicsController::class);
52 Route::apiResource('sections', SectionsController::class);
53 Route::apiResource('tasks', TasksController::class);
54 Route::apiResource('exam_tasks', Exam_tasksController::class);
55 Route::apiResource('model_doc_refs', Model_doc_refsController::class);
56 Route::apiResource('logbook_entries', Logbook_entriesController::class);
57 Route::apiResource('trainee_notes', Trainee_notesController::class);
58 Route::apiResource('trainer_notes', Trainer_notesController::class);
59 Route::apiResource('trainee_doc_refs', Trainee_doc_refsController::class);
60 Route::apiResource('evaluations', EvaluationsController::class);
61

```

Las rutas quedarán del siguiente modo:

```
C:\xampp\htdocs\aliaz> php artisan route:list
```

| Method    | URI   | Controller   | Actions          |
|-----------|---|--|------------------|
| GET HEAD  | /   | App\\Http\\Controllers\\Ignition\\IndexController              | index            |
| GET HEAD  | /ignite/healthcheck                                     | App\\Http\\Controllers\\Ignition\\HealthCheckController        | health           |
| POST      | /ignite/update-config                                   | App\\Http\\Controllers\\Ignition\\UpdateConfigController       | updateConfig     |
| POST      | /api/airports   | App\\Http\\Controllers\\Aircraft\\AircraftController           | store            |
| PUT PATCH | /api/airports/{aircraft}                                | App\\Http\\Controllers\\Aircraft\\AircraftController           | update           |
| DELETE    | /api/airports/{aircraft}                                | App\\Http\\Controllers\\Aircraft\\AircraftController           | destroy          |
| PUT PATCH | /api/chapters/{chapter}                                 | App\\Http\\Controllers\\Chapter\\ChapterController             | update           |
| DELETE    | /api/chapters/{chapter}                                 | App\\Http\\Controllers\\Chapter\\ChapterController             | destroy          |
| PUT PATCH | /api/evaluations/{evaluation}                           | App\\Http\\Controllers\\Evaluation\\EvaluationController       | update           |
| DELETE    | /api/evaluations/{evaluation}                           | App\\Http\\Controllers\\Evaluation\\EvaluationController       | destroy          |
| PUT PATCH | /api/exam_tasks/{exam_task}                             | App\\Http\\Controllers\\ExamTask\\ExamTaskController           | update           |
| DELETE    | /api/exam_tasks/{exam_task}                             | App\\Http\\Controllers\\ExamTask\\ExamTaskController           | destroy          |
| PUT PATCH | /api/exam_tasks/{exam_task}/update                      | App\\Http\\Controllers\\ExamTask\\ExamTaskController           | update           |
| PUT PATCH | /api/join_trainees/{logbook}                            | App\\Http\\Controllers\\JoinTrainee\\JoinTraineeController     | index            |
| POST      | /api/join_trainees/{logbook}                            | App\\Http\\Controllers\\JoinTrainee\\JoinTraineeController     | store            |
| PUT PATCH | /api/join_trainees/{logbook}/join_trainers/{logbook}    | App\\Http\\Controllers\\JoinTrainee\\JoinTraineeController     | update           |
| DELETE    | /api/join_trainees/{logbook}                            | App\\Http\\Controllers\\JoinTrainee\\JoinTraineeController     | destroy          |
| PUT PATCH | /api/join_trainers/{logbook}                            | App\\Http\\Controllers\\JoinTrainer\\JoinTrainerController     | index            |
| POST      | /api/join_trainers/{logbook}                            | App\\Http\\Controllers\\JoinTrainer\\JoinTrainerController     | store            |
| PUT PATCH | /api/join_trainers/{logbook}/join_trainees/{logbook}    | App\\Http\\Controllers\\JoinTrainer\\JoinTrainerController     | update           |
| DELETE    | /api/join_trainers/{logbook}                            | App\\Http\\Controllers\\JoinTrainer\\JoinTrainerController     | destroy          |
| PUT PATCH | /api/login_entries/{logbook_entry}                      | App\\Http\\Controllers\\LoginEntry\\LoginEntryController       | index            |
| POST      | /api/login_entries                                      | App\\Http\\Controllers\\LoginEntry\\LoginEntryController       | store            |
| PUT PATCH | /api/login_entries/{logbook_entry}/update               | App\\Http\\Controllers\\LoginEntry\\LoginEntryController       | update           |
| DELETE    | /api/login_entries/{logbook_entry}                      | App\\Http\\Controllers\\LoginEntry\\LoginEntryController       | destroy          |
| PUT PATCH | /api/logs/{logbook}                                     | App\\Http\\Controllers\\Log\\LogController                     | index            |
| POST      | /api/logs   | App\\Http\\Controllers\\Log\\LogController                     | store            |
| PUT PATCH | /api/logs/{logbook}                                     | App\\Http\\Controllers\\Log\\LogController                     | update           |
| DELETE    | /api/logs/{logbook}                                     | App\\Http\\Controllers\\Log\\LogController                     | destroy          |
| PUT PATCH | /api/logs/doc_refs/{model_doc_ref}                      | App\\Http\\Controllers\\LogDocRef\\LogDocRefController         | index            |
| POST      | /api/logs/doc_refs                                      | App\\Http\\Controllers\\LogDocRef\\LogDocRefController         | store            |
| PUT PATCH | /api/logs/doc_refs/{model_doc_ref}/update               | App\\Http\\Controllers\\LogDocRef\\LogDocRefController         | update           |
| DELETE    | /api/logs/doc_refs/{model_doc_ref}                      | App\\Http\\Controllers\\LogDocRef\\LogDocRefController         | destroy          |
| PUT PATCH | /api/products/{product}                                 | App\\Http\\Controllers\\Product\\ProductController             | index            |
| POST      | /api/products   | App\\Http\\Controllers\\Product\\ProductController             | store            |
| PUT PATCH | /api/products/{product}/update                          | App\\Http\\Controllers\\Product\\ProductController             | update           |
| DELETE    | /api/products/{product}                                 | App\\Http\\Controllers\\Product\\ProductController             | destroy          |
| PUT PATCH | /api/projects/{project}                                 | App\\Http\\Controllers\\Project\\ProjectController             | index            |
| POST      | /api/projects   | App\\Http\\Controllers\\Project\\ProjectController             | store            |
| PUT PATCH | /api/projects/{project}/update                          | App\\Http\\Controllers\\Project\\ProjectController             | update           |
| DELETE    | /api/projects/{project}                                 | App\\Http\\Controllers\\Project\\ProjectController             | destroy          |
| PUT PATCH | /api/sections/{section}                                 | App\\Http\\Controllers\\Section\\SectionController             | index            |
| POST      | /api/sections   | App\\Http\\Controllers\\Section\\SectionController             | store            |
| PUT PATCH | /api/specifications/{specification}                     | App\\Http\\Controllers\\Specification\\SpecificationController | index            |
| DELETE    | /api/specifications/{specification}                     | App\\Http\\Controllers\\Specification\\SpecificationController | destroy          |
| PUT PATCH | /api/tasks/{task}                                       | App\\Http\\Controllers\\Task\\TaskController                   | index            |
| DELETE    | /api/tasks/{task}                                       | App\\Http\\Controllers\\Task\\TaskController                   | destroy          |
| PUT PATCH | /api/topics/{topic}                                     | App\\Http\\Controllers\\Topic\\TopicController                 | index            |
| DELETE    | /api/topics/{topic}                                     | App\\Http\\Controllers\\Topic\\TopicController                 | destroy          |
| PUT PATCH | /api/topics/{topic}/update                              | App\\Http\\Controllers\\Topic\\TopicController                 | update           |
| PUT PATCH | /api/trainee_docs/{trainee_doc}/ref                     | App\\Http\\Controllers\\TraineeDoc\\TraineeDocController       | index            |
| POST      | /api/trainee_docs                                       | App\\Http\\Controllers\\TraineeDoc\\TraineeDocController       | store            |
| PUT PATCH | /api/trainee_docs/{trainee_doc}/ref_update              | App\\Http\\Controllers\\TraineeDoc\\TraineeDocController       | update           |
| DELETE    | /api/trainee_docs/{trainee_doc}/ref_destroy             | App\\Http\\Controllers\\TraineeDoc\\TraineeDocController       | destroy          |
| PUT PATCH | /api/trainee_notes/{trainee_note}/index                 | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | index            |
| POST      | /api/trainee_notes                                      | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | store            |
| PUT PATCH | /api/trainee_notes/{trainee_note}/update                | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | update           |
| DELETE    | /api/trainee_notes/{trainee_note}/destroy               | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | destroy          |
| PUT PATCH | /api/trainee_notes/{trainee_note}/ref_index             | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | index            |
| POST      | /api/trainee_notes/doc_refs                             | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | store            |
| PUT PATCH | /api/trainee_notes/doc_refs/{model_doc_ref}/ref_update  | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | update           |
| DELETE    | /api/trainee_notes/doc_refs/{model_doc_ref}/ref_destroy | App\\Http\\Controllers\\TraineeNote\\TraineeNoteController     | destroy          |
| PUT PATCH | /api/auth/login   | Auth\\Http\\Controllers\\AuthController                        | login            |
| POST      | /api/auth/logout  | Auth\\Http\\Controllers\\AuthController                        | logout           |
| POST      | /api/auth/confirm                                       | Auth\\Http\\Controllers\\AuthController                        | confirm          |
| POST      | /api/auth/password/confirm                              | Auth\\Http\\Controllers\\AuthController                        | passwordConfirm  |
| POST      | /api/auth/password/email                                | Auth\\Http\\Controllers\\AuthController                        | passwordEmail    |
| POST      | /api/auth/password/reset                                | Auth\\Http\\Controllers\\AuthController                        | passwordReset    |
| POST      | /api/auth/password/reset/{token}                        | Auth\\Http\\Controllers\\AuthController                        | passwordReset    |
| POST      | /api/auth/register                                      | Auth\\Http\\Controllers\\AuthController                        | register         |
| POST      | /api/auth/register/complete                             | Auth\\Http\\Controllers\\AuthController                        | registerComplete |
| GET HEAD  | /sanctum/csrf-cookie                                    | Laravel\\Sanctum\\CsrfCookieController                         | token            |

## Tarea 5 - Tests realizados con Postman.

En este apartado, se mostrarán todos los test correspondientes a cada una de las tablas, de forma que nos aseguramos que podemos hacer CRUD perfectamente en todas.

## Tarea 5.1 - Test de “aircraft”.

GET.

Overview GET http://localhost/elisa ● POST http://localhost/elisa ● PUT http://localhost/elisa ● DEL http://localhost/elisa ● + ⚙ No Environment

<http://localhost/elisa2/public/api/aircraft>

Save

Send

GET http://localhost/elisa2/public/api/aircraft

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

| KEY | VALUE | DESCRIPTION | ⋮ | Bulk Edit |
|-----|-------|-------------|---|-----------|
| Key | Value | Description |   |           |

Body Cookies Headers (11) Test Results

Status: 200 OK Time: 4.20 s Size: 699 B Save Response

Pretty Raw Preview Visualize JSON

```
1 
2 [
3   {
4     "id": 1,
5     "name": "A330",
6     "created_at": "2022-06-07T21:39:55.000000Z",
7     "updated_at": "2022-06-07T21:39:55.000000Z"
8   },
9   {
10     "id": 2,
11     "name": "A380",
12     "created_at": "2022-06-07T21:39:55.000000Z",
13     "updated_at": "2022-06-07T21:39:55.000000Z"
14   }
]
```

Runner Trash

POST.

The screenshot shows the Postman interface with the following details:

- URL:** GET http://localhost/elisa, POST http://localhost/elisa, PUT http://localhost/elisa, DELETE http://localhost/elisa
- Environment:** No Environment
- Request Type:** POST to http://localhost/elisa2/public/api/aircraft
- Body:** JSON (selected), Raw, Binary, GraphQL
- Params:** none, form-data, x-www-form-urlencoded
- Headers:** (10)
- Tests:** None
- Settings:** None
- Response:** Status: 200 OK, Time: 391 ms, Size: 488 B
- Raw Response Content:**

```
1 {  
2   "name": "Test aircraft"  
3 }
```



## UPDATE.

Overview GET http://localhost/elisa ● POST http://localhost/elisa ● PUT http://localhost/elisa ● DEL http://localhost/elisa ● + ... No Environment

http://localhost/elisa2/public/api/aircraft

POST http://localhost/elisa2/public/api/aircraft

Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Beautify

Body none form-data x-www-form-urlencoded raw binary GraphQL JSON

1  
2    "name": "Test aircraft"  
3

Body Cookies Headers (11) Test Results Status: 200 OK Time: 391 ms Size: 488 B Save Response

Pretty Raw Preview Visualize JSON

1  
2    "name": {  
3     "name": "Test aircraft",  
4     "updated\_at": "2022-06-13T09:52:13.000000Z",  
5     "created\_at": "2022-06-13T09:52:13.000000Z",  
6     "id": 5  
7 }

DELETE.

The screenshot shows the Postman application interface. At the top, there are several tabs: Overview, GET http://localhost/elisa, POST http://localhost/elisa, PUT http://localhost/elisa, and DEL http://localhost/elisa. The 'DEL' tab is currently selected. To the right of these tabs are buttons for '+', '...', and 'No Environment'. Below the tabs, there's a search bar with the URL 'http://localhost/elisa2/public/api/aircraft/5' and a 'Save' button with a dropdown menu. On the far right, there are two small icons for edit and copy.

The main workspace shows a 'DELETE' request with the URL 'http://localhost/elisa2/public/api/aircraft/5'. Below the request, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is selected. Under 'Query Params', there is a table with columns 'KEY', 'VALUE', 'DESCRIPTION', and 'Bulk Edit'. A single row is present with 'Key' as 'Value' and 'Description' as 'Description'. At the bottom of the table are 'Add' and 'Remove' buttons.

At the bottom of the interface, there are tabs for 'Body', 'Cookies', 'Headers (11)', and 'Test Results'. The 'Body' tab is selected. To the right of these tabs, there are status indicators: Status: 200 OK, Time: 166 ms, Size: 486 B, and a 'Save Response' button. Below the tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. There is also a small icon for expanding the JSON view.

The 'Pretty' tab is active, displaying the JSON response:

```
1  "name": {  
2      "id": 5,  
3      "name": "Update test",  
4      "created_at": "2022-06-13T09:52:13.000000Z",  
5      "updated_at": "2022-06-13T09:53:37.000000Z"  
6  }  
7 }
```



## Tarea 5.2 - Test de “specifics”.

GET.

Overview    GET http://localhost/elisa ●    POST http://localhost/elisa ●    PUT http://localhost/elisa ●    DEL http://localhost/elisa ●    +    ⚙    No Environment

http://localhost/elisa2/public/api/specfics

Send

GET    http://localhost/elisa2/public/api/specfics

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests    Settings    Cookies

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | ... |           |

Body    Cookies    Headers (11)    Test Results    Status: 200 OK    Time: 146 ms    Size: 1.03 KB    Save Response

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
235
236
237
237
238
239
239
240
241
242
243
244
244
245
246
246
247
247
248
248
249
249
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348

```

PUT.

Overview | GET http://localhost/elisa | POST http://localhost/elisa | PUT http://localhost/elisa | DEL http://localhost/elisa

<http://localhost/elisa2/public/api/specfics/7>

PUT http://localhost/elisa2/public/api/specfics/7

Params Authorization Headers (10) Body **JSON** Pre-request Script Tests Settings

Body Type: JSON

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 `{"name": "Update specifics"}`

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

1 `{"name": {`

2  `"id": 7,`

3  `"name": "Update specifics",`

4  `"created_at": "2022-06-13T10:00:00.000000Z",`

5  `"updated_at": "2022-06-13T10:11:19.000000Z"`

6 `}`

7

8

**DELETE.**

http://localhost/elisa2/public/api/specfics/7

DELETE ▾ http://localhost/elisa2/public/api/specfics/7

Params Authorization Headers (10) Body ● Pre-request Script Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ▾

```
1 [redacted]
2 "name": {
3     "id": 7,
4     "name": "Update specifics",
5     "created_at": "2022-06-13T0:10:00.000000Z",
6     "updated_at": "2022-06-13T0:11:19.000000Z"
7 }
```

## Tarea 5.3 - Test de “trainers”.

GET.

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elis](#) ● [PUT http://localhost/](#)

<http://localhost/elisa2/public/api/trainers>

GET    [http://localhost/elisa2/public/api/trainers](#)

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests    Settings

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

135
136
137
138
139
140
141
142
143
144
145
146
147
148
      "id": 23,
      "name": "trainer-23",
      "created_at": "2022-06-07T22:02:51.000000Z",
      "updated_at": "2022-06-07T22:02:51.000000Z"
    },
    {
      "id": 24,
      "name": "trainer-24",
      "created_at": "2022-06-07T22:02:51.000000Z",
      "updated_at": "2022-06-07T22:02:51.000000Z"
    }
  ]
}

```

POST.

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elis](#) ● [PUT h](#)

<http://localhost/elisa2/public/api/trainers>

POST    [http://localhost/elisa2/public/api/trainers](#)

Params    Authorization    Headers (10)    Body ●    Pre-request Script    Tests

none    form-data    x-www-form-urlencoded    raw    binary    Graph

1
2
3
 "name": "Trainer test"

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

1
2
3
      "name": {
        "name": "Trainer test",
        "updated_at": "2022-06-13T10:34:58.000000Z",
        "created_at": "2022-06-13T10:34:58.000000Z",
        "id": 25
      }
}

```

## PUT.

Overview | GET http://localhost/elisa | POST http://localhost/elisa | PUT http://localhost/elisa |

http://localhost/elisa2/public/api/trainers/25

PUT <http://localhost/elisa2/public/api/trainers/25>

Params Authorization Headers (10) Body **JSON** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   ...
3   "name": "Updated Trainer"
4 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "name": {
3     "id": 25,
4     "name": "Updated Trainer",
5     "created_at": "2022-06-13T10:34:58.000000Z",
6     "updated_at": "2022-06-13T10:36:07.000000Z"
7   }
8 }
```

## DELETE

Overview | GET http://localhost/elisa | POST http://localhost/elisa | PUT http://localhost/elisa | DEL http://localhost/elisa |

http://localhost/elisa2/public/api/trainers/25

DELETE <http://localhost/elisa2/public/api/trainers/25>

Params Authorization Headers (10) Body **JSON** Pre-request Script Tests Settings

Query Params

| KEY | VALUE | DESCRIPTION |
|-----|-------|-------------|
| Key | Value | Descriptio  |

Body Cookies Headers (11) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1 {
2   "name": {
3     "id": 25,
4     "name": "Updated Trainer",
5     "created_at": "2022-06-13T10:34:58.000000Z",
6     "updated_at": "2022-06-13T10:36:07.000000Z"
7   }
8 }
```

## Tarea 5.4 - Test de “trainees”.

### GET

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elisa](#) ● [PUT http://localhost/elisa](#)

[http://localhost/elisa2/public/api/trainees](#)

GET    [http://localhost/elisa2/public/api/trainees](#)

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests    Se

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

315   [
316     {
317       "id": 53,
318       "name": "trainee-53",
319       "created_at": "2022-06-07T22:05:19.000000Z",
320       "updated_at": "2022-06-07T22:05:19.000000Z"
321     },
322     [
323       {
324         "id": 54,
325         "name": "trainee-54",
326         "created_at": "2022-06-07T22:05:19.000000Z",
327         "updated_at": "2022-06-07T22:05:19.000000Z"
328     }
]

```

### POST

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elisa](#) ● [PUT http://localhost/elisa](#)

[http://localhost/elisa2/public/api/trainees](#)

POST    [http://localhost/elisa2/public/api/trainees](#)

Params    Authorization    Headers (10)    **Body** ●    Pre-request Script    Tests

none     form-data     x-www-form-urlencoded     raw     binary     Graph

```

1   [
2     {
3       "name": "Trainee test"
4     }
]

```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

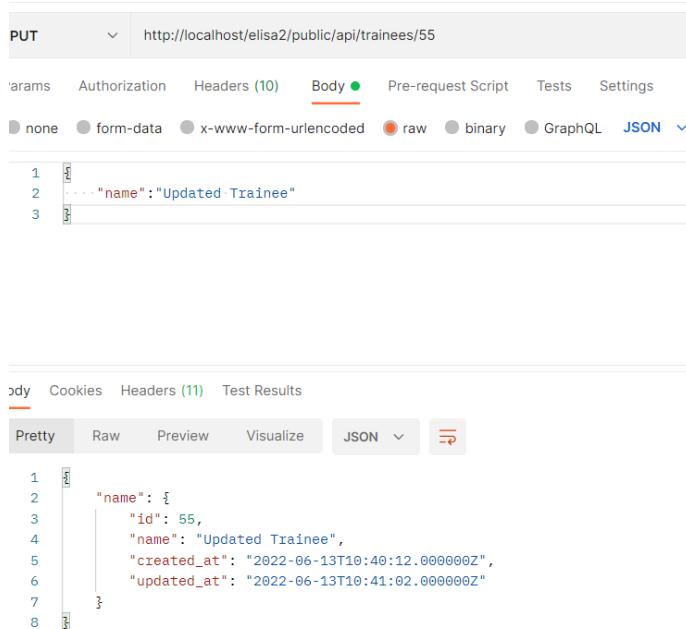
```

1   [
2     {
3       "name": "Trainee test"
4     }
]

```

## PUT

<http://localhost/elisa2/public/api/trainees/55>



PUT <http://localhost/elisa2/public/api/trainees/55>

Body `JSON`

```

1   "name": "Updated Trainee"
2
3
4
5
6
7
8
  
```

Pretty Raw Preview Visualize `JSON`

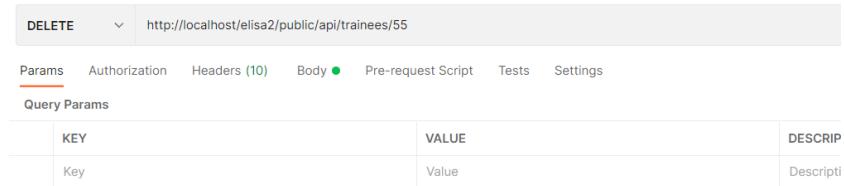
```

1
2   "name": {
3     "id": 55,
4     "name": "Updated Trainee",
5     "created_at": "2022-06-13T10:40:12.000000Z",
6     "updated_at": "2022-06-13T10:41:02.000000Z"
7
8
  
```

## DELETE

[Overview](#) | [GET http://localhost/elisa](#) | [POST http://localhost/elisa](#) | [PUT http://localhost/elisa](#) | [DEL http://localhost/elisa](#)

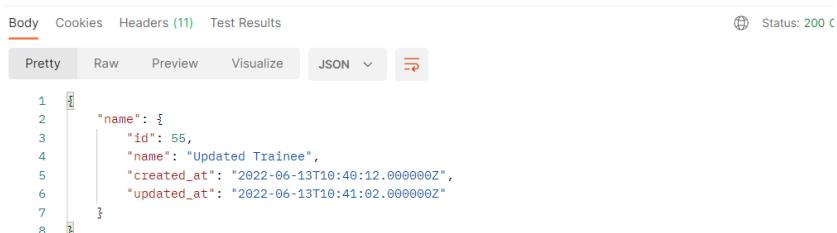
<http://localhost/elisa2/public/api/trainees/55>



DELETE <http://localhost/elisa2/public/api/trainees/55>

Query Params

| KEY | VALUE | DESCRIP  |
|-----|-------|----------|
| Key | Value | Descript |



Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize `JSON`

```

1
2   "name": {
3     "id": 55,
4     "name": "Updated Trainee",
5     "created_at": "2022-06-13T10:40:12.000000Z",
6     "updated_at": "2022-06-13T10:41:02.000000Z"
7
8
  
```

## Tarea 5.5 - Test de “chapters”.

### GET

Overview    GET http://localhost/elisa • POST http://localhost/elisa • PUT http://localhost/elisa

http://localhost/elisa2/public/api/chapters

GET    http://localhost/elisa2/public/api/chapters

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests    Settings

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

63   ],
64     "no": 18,
65     "name": "Example",
66     "created_at": "2022-06-09T17:44:45.00000Z",
67     "updated_at": "2022-06-09T17:44:45.00000Z"
68   },
69   {
70     "id": 5,
71     "no": 90,
72     "name": "Testing again",
73     "created_at": "2022-06-09T18:38:57.00000Z",
74     "updated_at": "2022-06-09T18:38:57.00000Z"
75   ]
76 }
```

### POST

Overview    GET http://localhost/elisa • POST http://localhost/elisa • PUT http://localhost/elisa

http://localhost/elisa2/public/api/chapters

POST    http://localhost/elisa2/public/api/chapters

Params    Authorization    Headers (10)    Body    Pre-request Script    Tests    Settings

None    Form-data    x-www-form-urlencoded    Raw    Binary    GraphQL

```

1   [
2     {
3       "no": 6,
4       "name": "Chapter test"
5     }
6   ]
```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

7   ],
8     "name": {
9       "no": 6,
10      "name": "Chapter test",
11      "updated_at": "2022-06-13T10:48:23.00000Z",
12      "created_at": "2022-06-13T10:48:23.00000Z",
13      "id": 6
14    }
15  ]
16 }
```

## PUT

Overview | GET http://localhost/elisa | POST http://localhost/elisa | PUT http://localhost/elisa | DEL http://localhost/elisa

**http://localhost/elisa2/public/api/chapters/6**

PUT http://localhost/elisa2/public/api/chapters/6

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```

1
2 ... "no": 20,
3 ... "name": "Chapter updated"
4

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

10
11     "id": 6,
12     "no": 20,
13     "name": "Chapter updated",
14     "created_at": "2022-06-13T10:48:23.000000Z",
15     "updated_at": "2022-06-13T10:50:03.000000Z"
16

```

## DELETE

Overview | GET http://localhost/elisa | POST http://localhost/elisa | PUT http://localhost/elisa | **DEL http://localhost/elisa**

**http://localhost/elisa2/public/api/chapters/6**

DELETE http://localhost/elisa2/public/api/chapters/6

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

Query Params

| KEY | VALUE | DESCRIP    |
|-----|-------|------------|
| Key | Value | Descriptio |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2     "no": {
3         "id": 6,
4         "no": 20,
5         "name": "Chapter updated",
6         "created_at": "2022-06-13T10:48:23.000000Z",
7         "updated_at": "2022-06-13T10:50:03.000000Z"

```

## Tarea 5.6 - Test de “products”.

### GET

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elis](#) ● [PUT http://localhost](#)

<http://localhost/elisa2/public/api/products>

GET    [http://localhost/elisa2/public/api/products](#)

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests    Settings

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    [JSON](#)

```

139     "name": "B737 CFM-56 B1 & B2 Practical Training",
140     "created_at": "2022-06-07T22:06:52.000000Z",
141     "updated_at": "2022-06-07T22:06:52.000000Z"
142   },
143   {
144     "id": 6,
145     "ac_id": 3,
146     "spec_id": 6,
147     "name": "B737 CFM-LEAP-1B B2 Practical Training",
148     "created_at": "2022-06-07T22:06:52.000000Z",
149     "updated_at": "2022-06-07T22:06:52.000000Z"
150   }
151 ]
152 }
```

### POST

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elis](#) ● [PUT h](#)

<http://localhost/elisa2/public/api/products>

POST    [http://localhost/elisa2/public/api/products](#)

Params    Authorization    Headers (10)    **Body** ●    Pre-request Script    Tests

none     form-data     x-www-form-urlencoded     raw     binary     GraphC

```

1
2   ...
3   ...
4   ...
5 }
```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    [JSON](#)

```

17
18   ...
19   ...
20   ...
21   ...
22   ...
23   ...
24   ...
25   ...
26 }
```

PUT

Overview    GET http://localhost/elisa ● POST http://localhost/elis ● PUT http://localhost/elisa ●

<http://localhost/elisa2/public/api/products/7>

PUT    http://localhost/elisa2/public/api/products/7

Params    Authorization    Headers (10)    Body    Pre-request Script    Tests    Settings

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL    JSON ▾

```
1    {  
2     "id": 7,  
3     "spec_id": 3,  
4     "name": "Product updated"  
5 }
```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON ▾

```
1 /  
2 {  
3   "name": {  
4     "id": 7,  
5     "ac_id": 2,  
6     "spec_id": 3,  
7     "name": "Product updated",  
8     "created_at": "2022-06-13T11:05:33.000000Z",  
9     "updated_at": "2022-06-13T11:06:30.000000Z"  
10   }  
11 }  
12 }
```

## DELETE

Overview

GET http://localhost/elisa ● POST http://localhost/elisa ● PUT

<http://localhost/elisa2/public/api/products/7>

**DELETE** ▾ http://localhost/elisa2/public/api/products/7

Params Authorization Headers (10) Body Pre-request Script Tests

Query Params

|  | KEY | VALUE |
|--|-----|-------|
|  | Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 [key]  
2 "ac_id": {  
3     "id": 7,  
4     "ac_id": 2,  
5     "spec_id": 3,  
6     "name": "Product updated",  
7     "created_at": "2022-06-13T11:05:33.000000Z",  
8     "updated_at": "2022-06-13T11:06:30.000000Z"  
9 },  
10 "spec_id": 5
```

## Tarea 5.7 - Test de “projects”.

### GET

Overview | GET http://localhost/elisa2/ | POST http://localhost/elisa2/ | PUT http://localhost/elisa2/ |

http://localhost/elisa2/public/api/projects

Method: GET | URL: http://localhost/elisa2/public/api/projects|

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

283     "name": "BT37 CFM-LEAP-1B B1 & B2 Practical Training Frankfurt Q3 2021",
284     "created_at": "2022-06-07T22:08:25.000000Z",
285     "updated_at": "2022-06-07T22:08:25.000000Z"
286   },
287   {
288     "id": 12,
289     "prod_id": 6,
290     "start_date": "2021-09-01",
291     "name": "BT37 CFM-LEAP-1B B1 & B2 Practical Training London Q3 2021",
292     "created_at": "2022-06-07T22:08:25.000000Z",
293     "updated_at": "2022-06-07T22:08:25.000000Z"
294   }
295 ]
296 
```

### POST

Overview | GET http://localhost/elisa2/ | POST http://localhost/elisa2/ |

http://localhost/elisa2/public/api/projects

Method: POST | URL: http://localhost/elisa2/public/api/projects|

Params Authorization Headers (10) Body Pre-request Script

Body: none form-data x-www-form-urlencoded raw binary

```

1   ...
2   ...
3   ...
4   ...
5   ... 
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

17   ...
18   ...
19     "name": {
20       "prod_id": 6,
21       "start_date": "2022-01-01",
22       "name": "Project test",
23       "updated_at": "2022-06-13T11:14:55.000000Z",
24       "created_at": "2022-06-13T11:14:55.000000Z",
25       "id": 13
26     } 
```

## PUT

Overview    GET http://localhost/elisa2/ ● POST http://localhost/elisa2/ ● PUT http://localhost/elisa2/public/api/projects/13

http://localhost/elisa2/public/api/projects/13

PUT    http://localhost/elisa2/public/api/projects/13

Params    Authorization    Headers (10)    **Body**    Pre-request Script    Tests    Settings

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL    **JSON**

```

1
2 ... "prod_id": 6,
3 ... "start_date": "2022-01-01",
4 ... "name": "Project update"
5

```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

1
2 ...
3   "prod_id": {
4     "id": 13,
5     "prod_id": 6,
6     "start_date": "2022-01-01",
7     "name": "Project update",
8     "created_at": "2022-06-13T11:14:55.000000Z",
9     "updated_at": "2022-06-13T11:24:03.000000Z"
10   },
11   "start_date": 5

```

## DELETE

DELETE    http://localhost/elisa2/public/api/projects/13

Params    Authorization    Headers (10)    Body    Pre-request Script    Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

1
2 ...
3   "prod_id": {
4     "id": 13,
5     "prod_id": 6,
6     "start_date": "2022-01-01",
7     "name": "Project update",
8     "created_at": "2022-06-13T11:14:55.000000Z",
9     "updated_at": "2022-06-13T11:24:03.000000Z"
10   },
11   "start_date": 5

```

## Tarea 5.8 - Test de “logbooks”.

### GET

Overview    GET http://localhost/elisa2/    POST http://localhost/elisa2/

http://localhost/elisa2/public/api/logbooks

GET    http://localhost/elisa2/public/api/logbooks

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

539     "proj_id": 9,
540     "name": "logbook 38 pi 9 tee 11 trs 3 4",
541     "created_at": "2022-06-07T22:11:13.000000Z",
542     "updated_at": "2022-06-07T22:11:13.000000Z"
543   },
544   {
545     "id": 39,
546     "proj_id": 9,
547     "name": "logbook 39 pi 9 tee 12 trs 3 4",
548     "created_at": "2022-06-07T22:11:13.000000Z",
549     "updated_at": "2022-06-07T22:11:13.000000Z"
550   }
551 }
552 }
```

### POST

http://localhost/elisa2/public/api/logbooks

POST    http://localhost/elisa2/public/api/logbooks

Params    Authorization    Headers (10)    Body    Pre-request Script

none    form-data    x-www-form-urlencoded    raw    binary

```

1
2   ...
3     "proj_id": 10,
4     "name": "logbook test"
5 }
```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

7     "id": 41
8   },
9   "name": {
10     "proj_id": 10,
11     "name": "logbook test",
12     "updated_at": "2022-06-13T11:28:38.000000Z",
13     "created_at": "2022-06-13T11:28:38.000000Z",
14     "id": 41
15   }
16 }
```

## PUT

<http://localhost/elisa2/public/api/logbooks/41>

PUT <http://localhost/elisa2/public/api/logbooks/41>

Params Authorization Headers (10) Body **Pre-request Script Tests**

none form-data x-www-form-urlencoded raw binary Graph

```

1
2   ...
3     "proj_id": 10,
4     "name": "logbook_update"
5
6
7
8
9
10
11
12
13
14
15
16

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

7   ...
8     "updated_at": "2022-06-13T11:29:27.000000Z",
9   },
10  "name": {
11    "id": 41,
12    "proj_id": 10,
13    "name": "logbook update",
14    "created_at": "2022-06-13T11:28:38.000000Z",
15    "updated_at": "2022-06-13T11:29:27.000000Z"
16  }

```

## DELETE

DELETE <http://localhost/elisa2/public/api/logbooks/41>

Params Authorization Headers (10) Body **Pre-request Script**

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   ...
3     "proj_id": {
4       "id": 41,
5       "proj_id": 10,
6       "name": "logbook update",
7       "created_at": "2022-06-13T11:28:38.000000Z",
8       "updated_at": "2022-06-13T11:29:27.000000Z"
9     },
10    "name": {
11      "id": 41,
12      "name": "logbook update"
13    }

```

## Tarea 5.9 - Test de “join\_trainers\_logbooks”.

### GET

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elis](#) ● [PUT http://localhost/elis](#)

[http://localhost/elisa2/public/api/join\\_trainers\\_logbooks](#)

GET    [http://localhost/elisa2/public/api/join\\_trainers\\_logbooks](#)

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests    S

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

1
2   "trainer_id": [
3     {
4       "id": 1,
5       "trainer_id": 1,
6       "logbook_id": 1,
7       "created_at": "2022-06-07T20:14:55.000000Z",
8       "updated_at": "2022-06-07T20:14:55.000000Z"
9     },
10    {
11      ...
12    }
13  ]
14
15
16

```

### POST

Overview    [GET http://localhost/elisa](#) ● [POST http://localhost/elis](#) ● [PUT http://localhost/elis](#)

[http://localhost/elisa2/public/api/join\\_trainers\\_logbooks](#)

POST    [http://localhost/elisa2/public/api/join\\_trainers\\_logbooks](#)

Params    Authorization    Headers (10)    **Body** ●    Pre-request Script    Tests    Settings

none     form-data     x-www-form-urlencoded     raw     binary     GraphQL

```

1
2   ...
3   ...
4   ...

```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

1
2   ...
3   ...
4   ...
5
6   ...
7   ...
8   ...
9   ...
10  ...
11  ...
12  ...
13  ...
14  ...
15  ...
16  ...

```

## PUT

Overview | [GET http://localhost/elisa2/](http://localhost/elisa2/) | [POST http://localhost/elisa2/](http://localhost/elisa2/)

[http://localhost/elisa2/public/api/join\\_trainers\\_logbooks/80](http://localhost/elisa2/public/api/join_trainers_logbooks/80)

PUT [http://localhost/elisa2/public/api/join\\_trainers\\_logbooks/80](http://localhost/elisa2/public/api/join_trainers_logbooks/80)

Params Authorization Headers (10) Body **Pre-request Script Test**

none  form-data  x-www-form-urlencoded  raw  binary  Graph

```

1
2 ... "trainer_id": 4,
3 ...
4 ...

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...

```

## DELETE

[http://localhost/elisa2/public/api/join\\_trainers\\_logbooks/80](http://localhost/elisa2/public/api/join_trainers_logbooks/80)

DELETE [http://localhost/elisa2/public/api/join\\_trainers\\_logbooks/80](http://localhost/elisa2/public/api/join_trainers_logbooks/80)

Params Authorization Headers (10) Body **Pre-request Script Test**

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...
9 ...
10 ...

```

## Tarea 5.10 - Test de “join\_trainees\_logbooks”.

### GET

http://localhost/elisa2/public/api/join\_trainees\_logbooks

GET http://localhost/elisa2/public/api/join\_trainees\_logbooks

Params Authorization Headers (6) Body Pre-request Script Tests Set

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "trainee_id": [
3     {
4       "id": 1,
5       "trainee_id": 1,
6       "logbook_id": 1,
7       "created_at": "2022-06-07T20:16:40.000000Z",
8       "updated_at": "2022-06-07T20:16:40.000000Z"
9     },
10    {
11      "id": 2,
12      "trainee_id": 2,
13      "logbook_id": 2,
14      "created_at": "2022-06-07T20:16:40.000000Z",
15      "updated_at": "2022-06-07T20:16:40.000000Z"
16    }
],

```

### POST

http://localhost/elisa2/public/api/join\_trainees\_logbooks

POST http://localhost/elisa2/public/api/join\_trainees\_logbooks

Params Authorization Headers (10) Body Pre-request Script 1

none form-data x-www-form-urlencoded raw binary

```

1 {
2   "trainee_id": 1,
3   "logbook_id": 1
4

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 42,
3   "trainee_id": 1,
4   "logbook_id": 1,
5   "updated_at": "2022-06-13T15:39:10.000000Z",
6   "created_at": "2022-06-13T15:39:10.000000Z",
7   "id": 42
},

```

## PUT

[http://localhost/elisa2/public/api/join\\_trainees\\_logbooks/42](http://localhost/elisa2/public/api/join_trainees_logbooks/42)

PUT [http://localhost/elisa2/public/api/join\\_trainees\\_logbooks/42](http://localhost/elisa2/public/api/join_trainees_logbooks/42)

Params Authorization Headers (10) Body  Pre-request Script

none  form-data  x-www-form-urlencoded  raw  binary

```

1
2 ....."trainee_id": 3,
3 ....."logbook_id": 3
4

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON 

```

1
2   "trainee_id": {
3     "id": 42,
4     "trainee_id": 3,
5     "logbook_id": 3,
6     "created_at": "2022-06-13T15:39:10.000000Z",
7     "updated_at": "2022-06-13T15:40:23.000000Z"
8   },
9   "logbook_id": {
10     "id": 42,
11     ".....": ...

```

## DELETE

[http://localhost/elisa2/public/api/join\\_trainees\\_logbooks/42](http://localhost/elisa2/public/api/join_trainees_logbooks/42)

DELETE [http://localhost/elisa2/public/api/join\\_trainees\\_logbooks/42](http://localhost/elisa2/public/api/join_trainees_logbooks/42)

Params Authorization Headers (10) Body  Pre-request Script Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON 

```

1
2   "trainee_id": {
3     "id": 42,
4     "trainee_id": 3,
5     "logbook_id": 3,
6     "created_at": "2022-06-13T15:39:10.000000Z",
7     "updated_at": "2022-06-13T15:40:23.000000Z"
8   },
9   "logbook_id": {
10     "id": 42,
11     ".....": ...

```

## Tarea 5.11 - Test de “topics”.

### GET

Overview    [GET http://localhost/elisa2/](http://localhost/elisa2/)    [POST http://localhost/elisa2/](http://localhost/elisa2/)

<http://localhost/elisa2/public/api/topics>

GET    [http://localhost/elisa2/public/api/topics](#)

Params    Authorization    Headers (6)    Body    Pre-request Script    Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

425 |   "chap_id": 3,
426 |   "no": 40,
427 |   "name": "topic 40 in chap 31",
428 |   "created_at": "2022-06-07T20:18:43.000000Z",
429 |   "updated_at": "2022-06-07T20:18:43.000000Z"
430 |
431 |
432 |   "id": 18,
433 |   "chap_id": 3,
434 |   "no": 50,
435 |   "name": "topic 50 in chap 31",
436 |   "created_at": "2022-06-07T20:18:43.000000Z",
437 |   "updated_at": "2022-06-07T20:18:43.000000Z"
438 |
439 ]
440 }
```

### POST

<http://localhost/elisa2/public/api/topics>

POST    [http://localhost/elisa2/public/api/topics](#)

Params    Authorization    Headers (10)    **Body**    Pre-request Script    Tests

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL

```

1   {
2     ...
3     "chap_id": 4,
4     "no": 100,
5     "name": "topics test"
6 }
```

Body    Cookies    Headers (11)    Test Results

Pretty    Raw    Preview    Visualize    JSON

```

454   },
455   {
456     "id": 19,
457     "chap_id": 4,
458     "no": 100,
459     "name": "topics test",
460     "created_at": "2022-06-13T16:38:26.000000Z",
461     "updated_at": "2022-06-13T16:38:26.000000Z"
462   }
463 ]
464 }
```

## PUT

<http://localhost/elisa2/public/api/topics/19>

PUT <http://localhost/elisa2/public/api/topics/19>

Params Authorization Headers (10) Body **Pre-request Script Tests Set**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

```

1
2 ..... "chap_id": 2,
3 ..... "no": 200,
4 ..... "name": "topics update"
5

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

```

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

## DELETE

<http://localhost/elisa2/public/api/topics/19>

PUT <http://localhost/elisa2/public/api/topics/19>

Params Authorization Headers (10) Body **Pre-request Script Tests Set**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

```

1
2 ..... "chap_id": 2,
3 ..... "no": 200,
4 ..... "name": "topics update"
5

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

```

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

## Tarea 5.12 - Test de “sections”.

GET

<http://localhost/elisa2/public/api/sections>

GET <http://localhost/elisa2/public/api/sections>

Params Authorization Headers (6) Body Pre-request Script Test

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

471   {
472     "id": 19,
473     "top_id": 8,
474     "no": 19,
475     "name": "sec 10 in topic 10 chap 24",
476     "created_at": "2022-06-07T20:20:03.000000Z",
477     "updated_at": "2022-06-07T20:20:03.000000Z"
478   },
479   {
480     "id": 20,
481     "top_id": 8,
482     "no": 20,
483     "name": "sec 20 in topic 10 chap 24",
484     "created_at": "2022-06-07T20:20:03.000000Z",
485     "updated_at": "2022-06-07T20:20:03.000000Z"
486 }
```

POST

<http://localhost/elisa2/public/api/sections>

POST <http://localhost/elisa2/public/api/sections>

Params Authorization Headers (10) Body Pre-request Script Test

Body

- none
- form-data
- x-www-form-urlencoded
- raw
- binary
- c

```

1   {
2     "top_id": 1,
3     "no": 1,
4     "name": "topic test"
5 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1   {
2     "top_id": {
3       "top_id": 1,
4       "no": 1,
5       "name": "topic test",
6       "updated_at": "2022-06-13T16:43:25.000000Z",
7       "created_at": "2022-06-13T16:43:25.000000Z",
8       "id": 21
9     },
10    "no": {
| }
```

## PUT

http://localhost/elisa2/public/api/sections/21

PUT http://localhost/elisa2/public/api/sections/21

Params   Authorization   Headers (10)   **Body** ●   Pre-request Script   Tests   Sett

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL

```

1
2 ... "top_id": 1,
3 ... "no": 1,
4 ... "name": "topic update"
5

```

Body   Cookies   Headers (11)   Test Results

Pretty   Raw   Preview   Visualize   JSON ✖

```

1
2 ...
3   "top_id": {
4     "id": 21,
5     "top_id": 1,
6     "no": 1,
7     "name": "topic update",
8     "created_at": "2022-06-13T16:43:25.000000Z",
9     "updated_at": "2022-06-13T16:44:05.000000Z"
10   },
11   "no": {
12     ...
13   }

```

## DELETE

http://localhost/elisa2/public/api/sections/21

DELETE http://localhost/elisa2/public/api/sections/21

Params   Authorization   Headers (10)   Body ●   Pre-request Script   Tes

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body   Cookies   Headers (11)   Test Results

Pretty   Raw   Preview   Visualize   JSON ✖

```

1
2 ...
3   "top_id": {
4     "id": 21,
5     "top_id": 1,
6     "no": 1,
7     "name": "topic update",
8     "created_at": "2022-06-13T16:43:25.000000Z",
9     "updated_at": "2022-06-13T16:44:05.000000Z"
10   },
11   "no": {
12     ...
13   }

```

## Tarea 5.13 - Test de “tasks”.

### GET

<http://localhost/elisa2/public/api/tasks>

GET <http://localhost/elisa2/public/api/tasks>

Params Authorization Headers (6) Body Pre-request Script Tests S

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

4033   "updated_at": "2022-06-07T20:23:19.000000Z"
4034   },
4035   {
4036     "id": 48,
4037     "prod_id": 6,
4038     "sect_id": 8,
4039     "ttype": "TS",
4040     "manob": "M",
4041     "level": 3,
4042     "is_exam": 0,
4043     "description": "sec 0 in topic 40 chap 31 prod 6",
4044     "created_at": "2022-06-07T20:23:19.000000Z",
4045     "updated_at": "2022-06-07T20:23:19.000000Z"
4046   }
4047 ]
4048

```

### POST

<http://localhost/elisa2/public/api/tasks>

POST <http://localhost/elisa2/public/api/tasks>

Params Authorization Headers (10) Body Pre-request Script Te

none form-data x-www-form-urlencoded raw binary C

```

1   {
2     "prod_id": 1,
3     "sect_id": 1,
4     "ttype": "LOC",
5     "manob": "O",
6     "level": 1,
7     "is_exam": 1,
8     "description": "task test"
9

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

75   "updated_at": "2022-06-13T16:51:26.000000Z",
76   "prod_id": 1,
77   "sect_id": 1,
78   "ttype": "LOC",
79   "manob": "O",
80   "level": 1,
81   "is_exam": 1,
82   "description": "task test",
83   "created_at": "2022-06-13T16:51:26.000000Z",
84   "id": 49
85

```

## PUT

<http://localhost/elisa2/public/api/tasks/49>

PUT [http://localhost/elisa2/public/api/tasks/49](#)

Params Authorization Headers (10) **Body** Pre-request Script 1

none form-data x-www-form-urlencoded raw binary

```

1
2     "prod_id": 2,
3     "sect_id": 2,
4     "ttype": "TS",
5     "manob": "M",
6     "level": 0,
7     "is_exam": 0,
8     "description": "task update"
9

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

75     "id": 49,
76     "prod_id": 2,
77     "sect_id": 2,
78     "ttype": "TS",
79     "manob": "M",
80     "level": 0,
81     "is_exam": 0,
82     "description": "task update",
83     "created_at": "2022-06-13T16:51:26.00000Z",
84     "updated_at": "2022-06-13T16:52:46.00000Z"
85
86

```

## DELETE

<http://localhost/elisa2/public/api/tasks/49>

DELETE [http://localhost/elisa2/public/api/tasks/49](#)

Params Authorization Headers (10) Body Pre-request Script

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

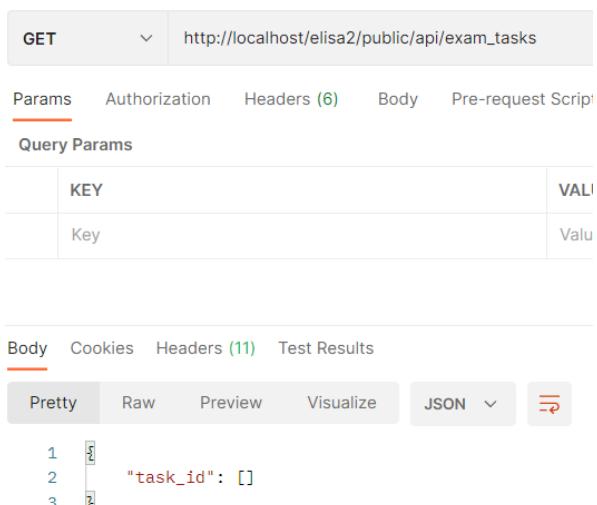
1
2     "prod_id": {
3         "id": 49,
4         "prod_id": 2,
5         "sect_id": 2,
6         "ttype": "TS",
7         "manob": "M",
8         "level": 0,
9         "is_exam": 0,
10        "description": "task update",
11        "created_at": "2022-06-13T16:51:26.00000Z",
12        "updated_at": "2022-06-13T16:52:46.00000Z"

```

## Tarea 5.14 - Test de “exam\_tasks”.

### GET

[http://localhost/elisa2/public/api/exam\\_tasks](http://localhost/elisa2/public/api/exam_tasks)



Postman screenshot showing a GET request to `http://localhost/elisa2/public/api/exam_tasks`. The request includes a query parameter `task_id` with value `1`.

| KEY | VAL   |
|-----|-------|
| Key | Value |

Response body:

```
1 [ ]  
2 "task_id": []  
3 [ ]
```

### POST

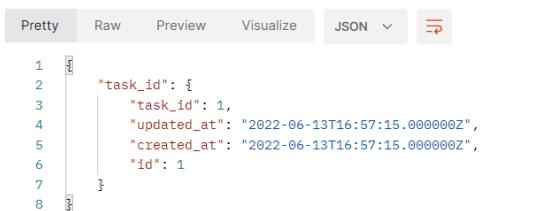
[http://localhost/elisa2/public/api/exam\\_tasks](http://localhost/elisa2/public/api/exam_tasks)



Postman screenshot showing a POST request to `http://localhost/elisa2/public/api/exam_tasks`. The request body contains a JSON object with `task_id` set to `1`.

```
1 [ ]  
2 "task_id": 1  
3 [ ]
```

Body Cookies Headers (11) Test Results



Postman screenshot showing the response body of the POST request. It is a JSON object with `task_id` containing another object with `task_id`, `updated_at`, `created_at`, and `id` fields.

```
1 [ ]  
2 "task_id": {  
3     "task_id": 1,  
4     "updated_at": "2022-06-13T16:57:15.000000Z",  
5     "created_at": "2022-06-13T16:57:15.000000Z",  
6     "id": 1  
7 }  
8 [ ]
```

## PUT

[http://localhost/elisa2/public/api/exam\\_tasks/1](http://localhost/elisa2/public/api/exam_tasks/1)

PUT [▼](#) http://localhost/elisa2/public/api/exam\_tasks/1

Params Authorization Headers (10) Body [●](#) Pre-request Script

none  form-data  x-www-form-urlencoded  raw  binary

```

1 {
2   ...
3 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON [▼](#) [≡](#)

```

1 "task_id": {
2   "id": 1,
3   "task_id": 2,
4   "created_at": "2022-06-13T16:57:15.000000Z",
5   "updated_at": "2022-06-13T16:57:48.000000Z"
6 }
```

## DELETE

[http://localhost/elisa2/public/api/exam\\_tasks/1](http://localhost/elisa2/public/api/exam_tasks/1)

DELETE [▼](#) http://localhost/elisa2/public/api/exam\_tasks/1

Params Authorization Headers (10) Body [●](#) Pre-request Script

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON [▼](#) [≡](#)

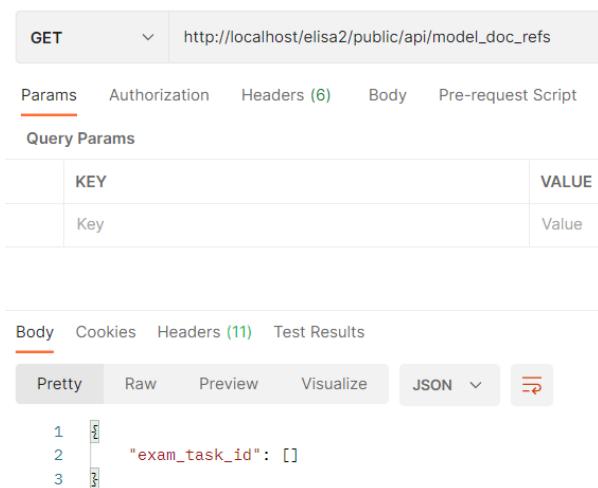
```

1 "task_id": {
2   "id": 1,
3   "task_id": 2,
4   "created_at": "2022-06-13T16:57:15.000000Z",
5   "updated_at": "2022-06-13T16:57:48.000000Z"
6 }
```

## Tarea 5.15 - Test de “model\_doc\_refs”.

### GET

[http://localhost/elisa2/public/api/model\\_doc\\_refs](http://localhost/elisa2/public/api/model_doc_refs)



GET [http://localhost/elisa2/public/api/model\\_doc\\_refs](http://localhost/elisa2/public/api/model_doc_refs)

Params Authorization Headers (6) Body Pre-request Script

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1
2 "exam_task_id": []
3
```

### POST

[http://localhost/elisa2/public/api/model\\_doc\\_refs](http://localhost/elisa2/public/api/model_doc_refs)



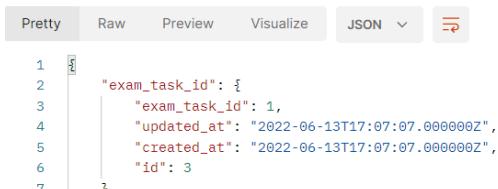
POST [http://localhost/elisa2/public/api/model\\_doc\\_refs](http://localhost/elisa2/public/api/model_doc_refs)

Params Authorization Headers (10) Body ● Pre-request Script

none form-data x-www-form-urlencoded raw binary

```
1
2 ... "exam_task_id": 1
3
```

Body Cookies Headers (11) Test Results



Pretty Raw Preview Visualize JSON ↻

```
1
2 "exam_task_id": {
3   "exam_task_id": 1,
4   "updated_at": "2022-06-13T17:07:07.00000Z",
5   "created_at": "2022-06-13T17:07:07.00000Z",
6   "id": 3
7 }
```

## PUT

[http://localhost/elisa2/public/api/model\\_doc\\_refs/1](http://localhost/elisa2/public/api/model_doc_refs/1)

PUT [http://localhost/elisa2/public/api/model\\_doc\\_refs/1](#)

Params Authorization Headers (10) Body **Pre-request Script Test**

none  form-data  x-www-form-urlencoded  raw  binary  G

```

1 {
2   "exam_task_id": 2
3 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON 

```

1 {
2   "exam_task_id": {
3     "id": 1,
4     "exam_task_id": 2,
5     "created_at": "2022-06-13T17:26:04.000000Z",
6     "updated_at": "2022-06-13T17:28:24.000000Z"
7   }
8 }
```

## DELETE

[http://localhost/elisa2/public/api/model\\_doc\\_refs/1](http://localhost/elisa2/public/api/model_doc_refs/1)

DELETE [http://localhost/elisa2/public/api/model\\_doc\\_refs/1](#)

Params Authorization Headers (10) Body **Pre-request Script**

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON 

```

1 {
2   "exam_task_id": {
3     "id": 1,
4     "exam_task_id": 2,
5     "created_at": "2022-06-13T17:26:04.000000Z",
6     "updated_at": "2022-06-13T17:28:24.000000Z"
7   }
8 }
```

## Tarea 5.16 - Test de “logbook\_entries”.

### GET

[http://localhost/elisa2/public/api/logbook\\_entries](http://localhost/elisa2/public/api/logbook_entries)

GET [http://localhost/elisa2/public/api/logbook\\_entries](http://localhost/elisa2/public/api/logbook_entries)

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

859     "tr_signoff": "0000-00-00 00:00:00",
860     "te_signoff": "0000-00-00 00:00:00",
861     "created_at": "2022-06-07T18:50:59.000000Z",
862     "updated_at": "2022-06-07T18:50:59.000000Z"
863   },
864   {
865     "id": 24,
866     "logbook_id": 37,
867     "task_id": 40,
868     "tr_signoff": "0000-00-00 00:00:00",
869     "te_signoff": "0000-00-00 00:00:00",
870     "created_at": "2022-06-07T18:50:59.000000Z",
871     "updated_at": "2022-06-07T18:50:59.000000Z"
872   }
873 ]
874 }
```

### POST

[http://localhost/elisa2/public/api/logbook\\_entries](http://localhost/elisa2/public/api/logbook_entries)

POST [http://localhost/elisa2/public/api/logbook\\_entries](http://localhost/elisa2/public/api/logbook_entries)

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

```

1
2     "logbook_id": 1,
3     "task_id": 1,
4     "tr_signoff": "2022-01-01 00:00:00",
5     "te_signoff": "2022-01-01 00:00:00"
6 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

27     "id": 25
28   },
29   "te_signoff": {
30     "logbook_id": 1,
31     "task_id": 1,
32     "tr_signoff": "2022-01-01 00:00:00",
33     "te_signoff": "2022-01-01 00:00:00",
34     "updated_at": "2022-06-13T17:33:45.000000Z",
35     "created_at": "2022-06-13T17:33:45.000000Z",
36   }
37 }
```

## PUT

[http://localhost/elisa2/public/api/logbook\\_entries/25](http://localhost/elisa2/public/api/logbook_entries/25)

PUT [http://localhost/elisa2/public/api/logbook\\_entries/25](http://localhost/elisa2/public/api/logbook_entries/25)

Params Authorization Headers (10) **Body** Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary GraphQL

```

1
2     "logbook_id": 3,
3     "task_id": 3,
4     "tr_signoff": "2022-01-03 00:00:00",
5     "te_signoff": "2022-01-02 00:00:00"
6

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

27     "updated_at": "2022-06-13T17:35:57.00000Z"
28
29     "te_signoff": {
30         "id": 25,
31         "logbook_id": 3,
32         "task_id": 3,
33         "tr_signoff": "2022-01-03 00:00:00",
34         "te_signoff": "2022-01-02 00:00:00",
35         "created_at": "2022-06-13T17:33:45.00000Z",
36         "updated_at": "2022-06-13T17:35:57.00000Z"
37     }
38

```

## DELETE

[http://localhost/elisa2/public/api/logbook\\_entries/25](http://localhost/elisa2/public/api/logbook_entries/25)

DELETE [http://localhost/elisa2/public/api/logbook\\_entries/25](http://localhost/elisa2/public/api/logbook_entries/25)

Params Authorization Headers (10) Body Pre-request Script Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2     "logbook_id": {
3         "id": 25,
4         "logbook_id": 3,
5         "task_id": 3,
6         "tr_signoff": "2022-01-03 00:00:00",
7         "te_signoff": "2022-01-02 00:00:00",
8         "created_at": "2022-06-13T17:33:45.00000Z",
9         "updated_at": "2022-06-13T17:35:57.00000Z"
10    },
11    "task_id": {
12        "id": 25,
13    }

```

## Tarea 5.17 - Test de “trainee\_notes”.

### GET

[http://localhost/elisa2/public/api/trainee\\_notes](http://localhost/elisa2/public/api/trainee_notes)

GET [http://localhost/elisa2/public/api/trainee\\_notes](http://localhost/elisa2/public/api/trainee_notes)

Params Authorization Headers (6) Body Pre-request Script Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 {
2   "lb_entry_id": [
3     {
4       "id": 3,
5       "lb_entry_id": 10,
6       "created_at": "2022-06-07T18:52:29.000000Z",
7       "updated_at": "2022-06-07T18:52:29.000000Z"
8     },
9     {
10       "id": 4,
11       "lb_entry_id": 12,
12       "created_at": "2022-06-07T18:52:29.000000Z",
13       "updated_at": "2022-06-07T18:52:29.000000Z"
14     }
15   ]
16 }
```

### POST

[http://localhost/elisa2/public/api/trainee\\_notes](http://localhost/elisa2/public/api/trainee_notes)

POST [http://localhost/elisa2/public/api/trainee\\_notes](http://localhost/elisa2/public/api/trainee_notes)

Params Authorization Headers (10) Body Pre-request Script Tests ⚙

none form-data x-www-form-urlencoded raw binary GraphQL

```

1 {
2   ...
3 }
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 {
2   "lb_entry_id": {
3     "lb_entry_id": 1,
4     "updated_at": "2022-06-13T17:46:36.000000Z",
5     "created_at": "2022-06-13T17:46:36.000000Z",
6     "id": 5
7   }
8 }
```

## PUT

[http://localhost/elisa2/public/api/trainee\\_notes/5](http://localhost/elisa2/public/api/trainee_notes/5)

PUT [http://localhost/elisa2/public/api/trainee\\_notes/5](http://localhost/elisa2/public/api/trainee_notes/5)

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```

1 ...
2 ... "lb_entry_id": 2
3

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1 ...
2 "lb_entry_id": {
3     "id": 5,
4     "lb_entry_id": 2,
5     "created_at": "2022-06-13T17:46:36.00000Z",
6     "updated_at": "2022-06-13T17:46:56.00000Z"
7 }
8

```

## DELETE

[http://localhost/elisa2/public/api/trainee\\_notes/5](http://localhost/elisa2/public/api/trainee_notes/5)

DELETE [http://localhost/elisa2/public/api/trainee\\_notes/5](http://localhost/elisa2/public/api/trainee_notes/5)

Params Authorization Headers (10) Body Pre-request Script Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1 ...
2 "lb_entry_id": {
3     "id": 5,
4     "lb_entry_id": 2,
5     "created_at": "2022-06-13T17:46:36.00000Z",
6     "updated_at": "2022-06-13T17:46:56.00000Z"
7 }
8

```

## Tarea 5.18 - Test de “trainer\_notes”.

GET

[http://localhost/elisa2/public/api/trainer\\_notes](http://localhost/elisa2/public/api/trainer_notes)

GET [http://localhost/elisa2/public/api/trainer\\_notes](http://localhost/elisa2/public/api/trainer_notes)

Params Authorization Headers (6) Body Pre-request Script Test

Query Params

|  | KEY | VALUE |
|--|-----|-------|
|  | Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 [
2   "lb_entry_id": []
3 ]

```

POST

[http://localhost/elisa2/public/api/trainer\\_notes](http://localhost/elisa2/public/api/trainer_notes)

POST [http://localhost/elisa2/public/api/trainer\\_notes](http://localhost/elisa2/public/api/trainer_notes)

Params Authorization Headers (10) Body  Pre-request Script

none form-data x-www-form-urlencoded raw binary

```

1 {
2   "lb_entry_id": 1
3 }

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 {
2   "lb_entry_id": {
3     "lb_entry_id": 1,
4     "updated_at": "2022-06-13T17:49:23.00000Z",
5     "created_at": "2022-06-13T17:49:23.00000Z",
6     "id": 1
7   }
8 }

```

## PUT

[http://localhost/elisa2/public/api/trainer\\_notes/1](http://localhost/elisa2/public/api/trainer_notes/1)

PUT [http://localhost/elisa2/public/api/trainer\\_notes/1](http://localhost/elisa2/public/api/trainer_notes/1)

Params Authorization Headers (10) Body **Pre-request Script** T

Body  form-data  x-www-form-urlencoded  raw  binary

```

1
2 ... "lb_entry_id": 2
3

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON 

```

1
2   "lb_entry_id": {
3     "id": 1,
4     "lb_entry_id": 2,
5     "created_at": "2022-06-13T17:49:23.000000Z",
6     "updated_at": "2022-06-13T17:49:36.000000Z"
7   }
8

```

## DELETE

[http://localhost/elisa2/public/api/trainer\\_notes/1](http://localhost/elisa2/public/api/trainer_notes/1)

DELETE [http://localhost/elisa2/public/api/trainer\\_notes/1](http://localhost/elisa2/public/api/trainer_notes/1)

Params Authorization Headers (10) Body **Pre-request Script** Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON 

```

1
2   "lb_entry_id": {
3     "id": 1,
4     "lb_entry_id": 2,
5     "created_at": "2022-06-13T17:49:23.000000Z",
6     "updated_at": "2022-06-13T17:49:36.000000Z"
7   }
8

```

## Tarea 5.19 - Test de “trainee\_doc\_refs”.

### GET

[http://localhost/elisa2/public/api/trainee\\_doc\\_refs](http://localhost/elisa2/public/api/trainee_doc_refs)

GET [http://localhost/elisa2/public/api/trainee\\_doc\\_refs](http://localhost/elisa2/public/api/trainee_doc_refs)

Params Authorization Headers (6) Body Pre-request Script

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 "lb_entry_id": []
2
3

```

### POST

[http://localhost/elisa2/public/api/trainee\\_doc\\_refs](http://localhost/elisa2/public/api/trainee_doc_refs)

POST [http://localhost/elisa2/public/api/trainee\\_doc\\_refs](http://localhost/elisa2/public/api/trainee_doc_refs)

Params Authorization Headers (10) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary ↻

```

1 ...
2 ... "lb_entry_id": 1
3

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 "lb_entry_id": {
2   "lb_entry_id": 1,
3   "updated_at": "2022-06-13T17:52:10.000000Z",
4   "created_at": "2022-06-13T17:52:10.000000Z",
5   "id": 1
6 }
7
8

```

## PUT

[http://localhost/elisa2/public/api/trainee\\_doc\\_refs/1](http://localhost/elisa2/public/api/trainee_doc_refs/1)

PUT [http://localhost/elisa2/public/api/trainee\\_doc\\_refs/1](http://localhost/elisa2/public/api/trainee_doc_refs/1)

Params Authorization Headers (10) Body **raw** Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary Graph

```

1
2 ... "lb_entry_id": 2
3

```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "lb_entry_id": {
3     "id": 1,
4     "lb_entry_id": 2,
5     "created_at": "2022-06-13T17:52:10.000000Z",
6     "updated_at": "2022-06-13T17:52:28.000000Z"
7   }
8

```

## DELETE

[http://localhost/elisa2/public/api/trainee\\_doc\\_refs/1](http://localhost/elisa2/public/api/trainee_doc_refs/1)

DELETE [http://localhost/elisa2/public/api/trainee\\_doc\\_refs/1](http://localhost/elisa2/public/api/trainee_doc_refs/1)

Params Authorization Headers (10) Body **raw** Pre-request Script Tests

Query Params

| KEY | VALUE |
|-----|-------|
| Key | Value |

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```

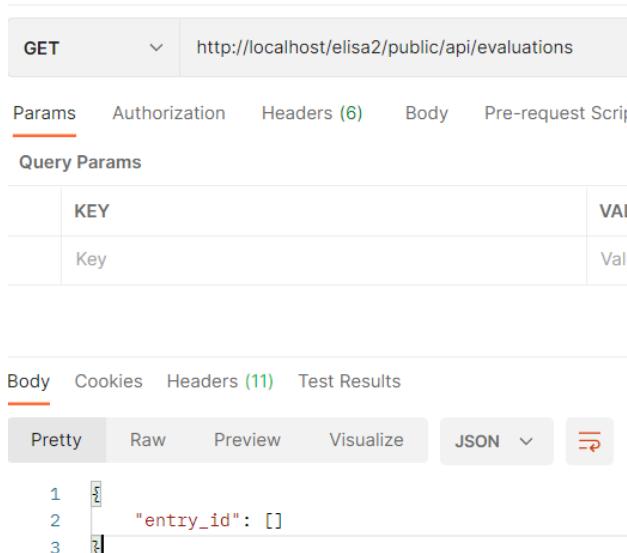
1
2   "lb_entry_id": {
3     "id": 1,
4     "lb_entry_id": 2,
5     "created_at": "2022-06-13T17:52:10.000000Z",
6     "updated_at": "2022-06-13T17:52:28.000000Z"
7   }
8

```

## Tarea 5.20 - Test de “evaluations”.

GET

<http://localhost/elisa2/public/api/evaluations>



Postman screenshot showing a GET request to `http://localhost/elisa2/public/api/evaluations`. The request includes a query parameter `Key`. The response body is shown in Pretty JSON format:

```
1 [ {  
2   "entry_id": []  
3 } ]
```

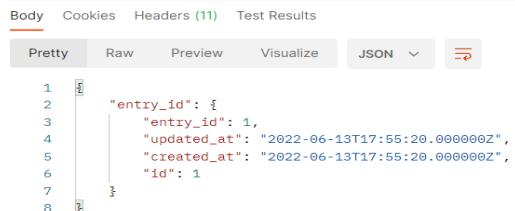
POST

<http://localhost/elisa2/public/api/evaluations>



Postman screenshot showing a POST request to `http://localhost/elisa2/public/api/evaluations`. The request body contains a single entry with `entry_id: 1`. The response body is shown in Pretty JSON format:

```
1 [ {  
2   "entry_id": 1  
3 } ]
```



Postman screenshot showing a POST request to `http://localhost/elisa2/public/api/evaluations`. The request body contains a single entry with `entry_id: 1`. The response body is shown in Pretty JSON format:

```
1 [ {  
2   "entry_id": 1,  
3   "updated_at": "2022-06-13T17:55:20.000000Z",  
4   "created_at": "2022-06-13T17:55:20.000000Z",  
5   "id": 1  
6 } ]
```

## PUT

<http://localhost/elisa2/public/api/evaluations/1>

PUT <http://localhost/elisa2/public/api/evaluations/1>

Params Authorization Headers (10) Body **Pre-request Script**

none form-data x-www-form-urlencoded raw binary

```
1 ... "entry_id": 2
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "entry_id": {
3     "id": 1,
4     "entry_id": 2,
5     "created_at": "2022-06-13T17:55:20.000000Z",
6     "updated_at": "2022-06-13T17:55:26.000000Z"
7 }
```

## DELETE

<http://localhost/elisa2/public/api/evaluations/1>

DELETE <http://localhost/elisa2/public/api/evaluations/1>

Params Authorization Headers (10) Body **Pre-request Script** Tests

none form-data x-www-form-urlencoded raw binary Graph

```
1 ... "entry_id": 2
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "entry_id": {
3     "id": 1,
4     "entry_id": 2,
5     "created_at": "2022-06-13T17:55:20.000000Z",
6     "updated_at": "2022-06-13T17:55:26.000000Z"
7 }
```



123

## Tarea 6 - Configuración inicial de autenticación.

La autenticación en Laravel es un proceso que nos permite comprobar que el usuario es quien dice ser, por medio de un nombre de usuario o una dirección de e-mail y una contraseña.

Este comando nos instala los elementos necesarios para la autenticación:

*composer require laravel/ui*

Para crear los elementos de autenticación, ejecutamos el comando:

*php artisan ui bootstrap --auth*

Esto creará de manera automática mediante Bootstrap, una interfaz inicial de autenticación de usuarios.

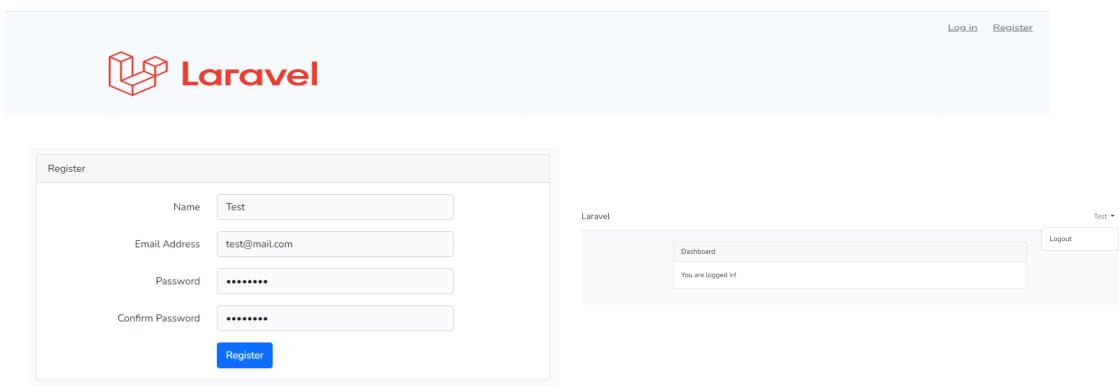
Por último, hay que hacer la instalación de Node.js, desde su página web. Una vez realizado, ejecutamos, por separado, los siguientes comandos:

*npm install*

*npm run dev*

Si nos lanza una advertencia, ejecutamos lo siguiente: *npm install autoprefixer@10.4.5 --save-exact*

Al acceder a nuestro localhost desde el navegador, nos carga la interfaz de autenticación, donde podemos logear y registrar nuevos usuarios.



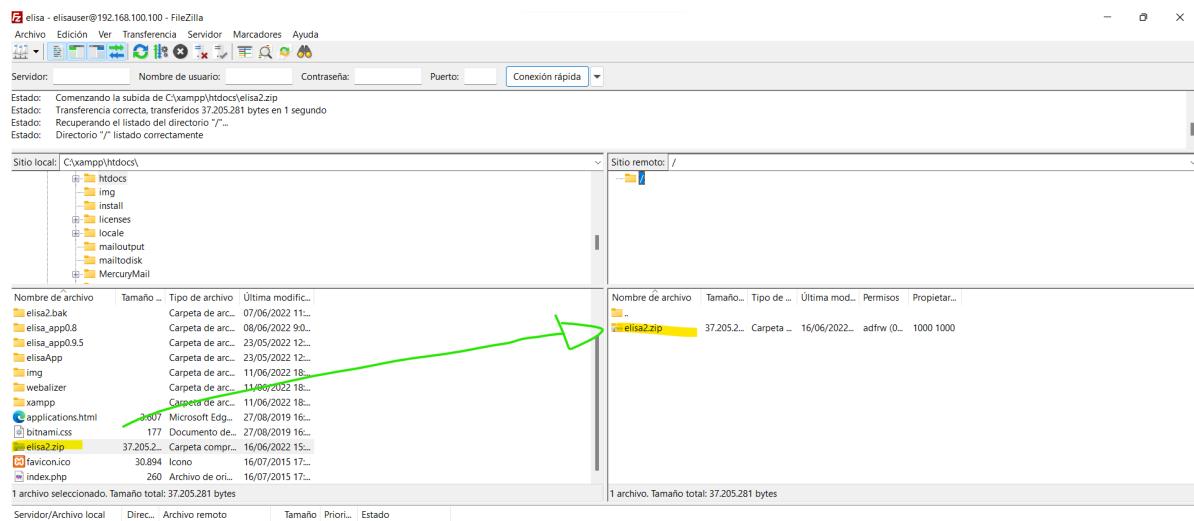
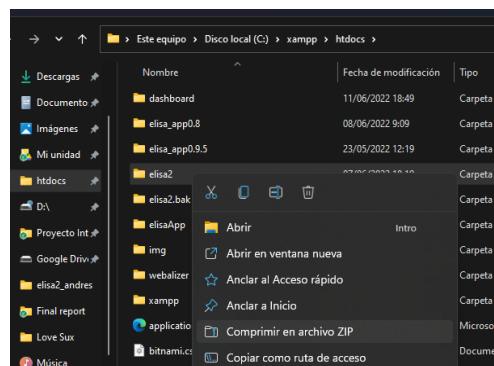
The screenshot shows a Laravel application interface. On the left, there is a 'Register' form with fields for Name (Test), Email Address (test@mail.com), Password, and Confirm Password. A 'Register' button is at the bottom. On the right, there is a 'Laravel' dashboard with a 'Dashboard' link and a message 'You are logged in!'. At the top right, there are 'Log in' and 'Register' links, and a dropdown menu showing 'Test'.

## Tarea 7 - Subida de la aplicación al servidor.

Para esta tarea, utilizaremos Filezilla, que dejamos configurado y listo en los pasos anteriores.

Vamos a nuestro sistema de archivos local y comprimimos el proyecto en un zip.

Y en Filezilla, arrastramos y pegamos el zip en la raíz de nuestro servidor.



Luego, en nuestro servidor, accedemos al recurso y descomprimimos con

unzip elisa2

```
inflating: elisa2/Webpack.mix.js
[elisauser@elisa2vm elisa2]$ s
bash: s: command not found
[elisauser@elisa2vm elisa2]$ ls
elisa2  elisa2.zip
[elisauser@elisa2vm elisa2]$ |
```

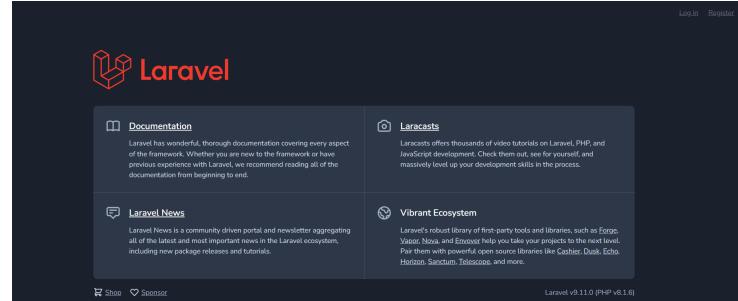
Este se descomprimirá en nuestro directorio dentro de una carpeta llamada elisa2.

Con Filezilla, podremos mover esos archivos a la carpeta inicial y así poder ejecutar sin problemas la configuración establecida.

Probamos que funciona poniendo en el navegador la ip local:

En el caso que no se muestre, cambiamos los permiso del directorio co:

*chmod -R 777 elisauser*



Ahora le toca el turno a la base de datos. Accederemos con root a MySQL y crearemos un usuario y una base de datos. Ese usuario accederá a todas las tablas de la base de datos “elisa2”.

```
CREATE DATABASE elisa2;
CREATE USER 'elisauser'@'localhost' IDENTIFIED BY 'elisaPass123';
GRANT ALL PRIVILEGES ON * . * TO 'elisauser'@'localhost';
FLUSH PRIVILEGES;
```

```
[root@elisa2vm html]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 16
Server version: 10.4.25-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE elisa2;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'elisauser'@'localhost' IDENTIFIED BY 'elisaPass123';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON * . * TO 'elisauser'@'localhost';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'elisauser'@'localhost';
+-----+
| Grants for elisauser@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO `elisauser`@`localhost` IDENTIFIED BY PASSWORD '*0851B2988B16934BC4B96A829E64552FAFCFB895' |
2 rows in set (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'PRIVILEGES' at line 1
MariaDB [(none)]>
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit;
Bye
```

Una ves logueados con elisauser, comprobamos que todo esté correcto.

Toca hacer las migraciones desde línea de comandos. Vamos al directorio raíz y editamos el archivo .env para poder realizar las migraciones.

Tenemos que sustituir las credenciales antiguas por las nuevas.

Y una vez hecho guardamos.

Al haber cambiado esta información, debemos correr los siguientes comandos para limpiar la caché y que no nos den problemas las migraciones:

```
php artisan cache:clear  
php artisan config:cache  
php artisan config:clear
```

Entonces es el momento de realizar las migraciones con el comando :

```
php artisan migrate
```

```
[elisauser@elisa2vm elisa2]$ php artisan cache:clear  
application cache cleared successfully.  
[elisauser@elisa2vm elisa2]$ php artisan config:cache  
configuration cache cleared successfully.  
configuration cached successfully.  
[elisauser@elisa2vm elisa2]$ php artisan config:clear  
configuration cache cleared successfully.  
[elisauser@elisa2vm elisa2]$ php artisan migrate  
migration table created successfully.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table (31,411.31ms)  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table (2,201.29ms)  
Migrating: 2019_08_19_000000_create_failed_jobs_table  
Migrated: 2019_08_19_000000_create_failed_jobs_table (8,900.06ms)  
Migrating: 2019_12_14_000001_create_personal_access_tokens_table  
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (1,874.60ms)  
Migrating: 2022_05_09_112702_create_aircraft_table
```

Y luego, con MySQL, podremos insertar todos los registros.

```

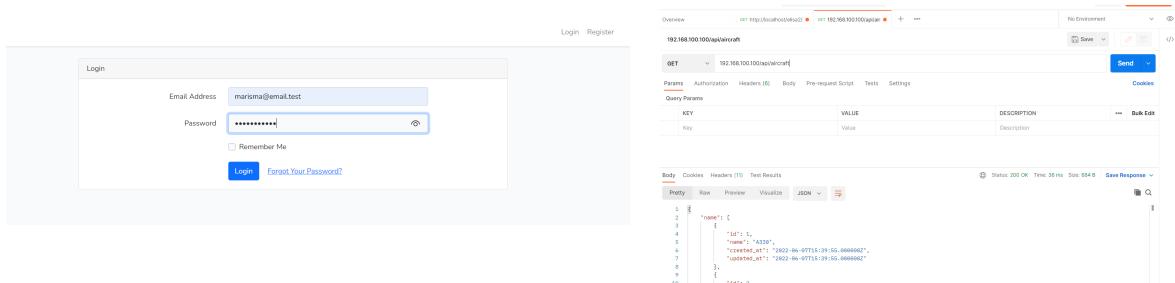
MariaDB [(none)]> USE DATABASE elisa2;
ERROR 1049 (42000): Unknown database 'DATABASE'
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| elisa2   |
| information_schema |
| mysql     |
| performance_schema |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> use elisa2
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [elisa2]> INSERT INTO `aircraft` ('id', 'name', 'created_at', 'updated_at') VALUES
    -> (1, 'A330', '2022-06-07 15:39:55', '2022-06-07 15:39:55'),
    -> (2, 'A380', '2022-06-07 15:39:55', '2022-06-07 15:39:55'),
    -> (3, 'B737', '2022-06-07 15:39:55', '2022-06-07 15:39:55');
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

```

Una vez realizado, tanto la página de inicio de sesión como la API funcionan correctamente bajo la IP pública de la red local.



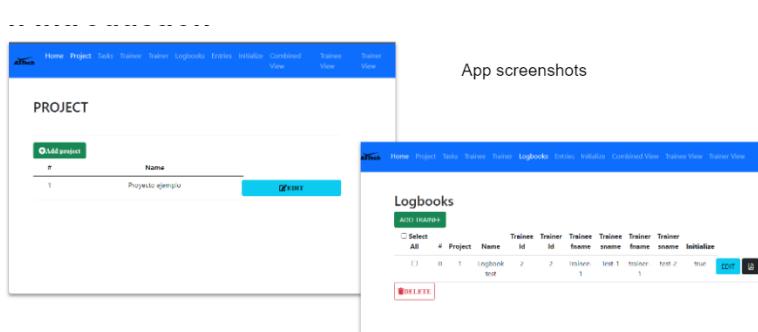
The screenshot shows a login interface on the left and a Postman API request on the right. The login form has fields for Email Address (marisma@email.test) and Password (\*\*\*\*\*). The Postman request is a GET to `http://192.168.100.100/api/aircraft`. The response body is a JSON array of aircraft objects:

```

[{"id": 1, "name": "A330", "created_at": "2022-06-07 15:39:55", "updated_at": "2022-06-07 15:39:55"}, {"id": 2, "name": "A380", "created_at": "2022-06-07 15:39:55", "updated_at": "2022-06-07 15:39:55"}, {"id": 3, "name": "B737", "created_at": "2022-06-07 15:39:55", "updated_at": "2022-06-07 15:39:55"}]

```

Ahora llega el turno del desarrollador front-end para empezar a implementar la API en el proyecto “Elisa”.



The screenshot displays two main sections of the Elisa application. On the left, the 'PROJECT' section shows a table with one row labeled 'Proyecto ejemplo'. On the right, the 'Logbooks' section shows a table with columns: Select, Project, Name, Trainee\_id, Trainer\_id, Trainee\_name, Trainer\_name, and Initialize. There is one entry: 'Logbook test'.

## 6. RECURSOS HUMANOS.

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

## 6. Recursos humanos.

En este apartado se definirán los perfiles de los recursos necesarios para la ejecución del proyecto y su participación en las tareas.

Para este proyecto, estaría implicado un administrador de sistemas con conocimientos en programación suficientes para llevar a cabo la puesta en marcha y creación de la API. Su salarios oscilaría entre 1800 euros y 2200 euros según la experiencia que posea y su trayectoria.

Se encargará en su mayoría, de las tareas referentes al back-end:

- Creación y configuración del servidor.
- Administración de permisos.
- Configuración del entorno de red.
- Diseño e implementación de la base de datos relacional.
- Instalación y configuración de los programas y servicios necesarios.
- Instalación y configuración de Laravel.
- Creación y test de la API.

Su trabajo irá en consonancia con el de un programador web al que dará soporte y apoyo de cara al desarrollo del prototipo.

Se estima una duración aproximada de un mes (1) para la realización del proyecto API.

## 7. RECURSOS MATERIALES.

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

## 7. Recursos materiales.

Se detallarán los recursos materiales necesarios para la ejecución o el éxito del proyecto.

- Estación de trabajo DELL Precision 3260 Compact Form Factor, 1 unidad, 1.747,25 €, Enlace → [Estación de trabajo Precision 3260 Compact | Dell España](#)
- Monitor profesional LG, dos unidades, 169,43 €/unidad, total: 338,86 €, → Enlace [Comprar Monitor LG IPS 24" - Tienda LG \(tiendalgonline.com\)](#)
- Teclado y ratón HP, 69,99 €, Enlace → [Teclado y ratón inalámbricos HP Pavilion 800 \(negro\) - HP Store España](#)
- Webcam Logitech Full HD, 110,80 €, Enlace → [Logitech StreamCam, Cámara Web con USB-C para Streaming de vídeo y creación de Contenido, Vídeo vertical Full HD 1080p a 60 fps, Versatilidad demontaje, para YouTube, Gaming Twitch, PC/Mac, Negro : Logitech: Amazon.es: Electrónica](#)
- Headset Logitech, 32,61 €, Enlace → [Logitech H390 USB Headset : Amazon.es: Informática](#)

El coste total del equipamiento necesario para poder llevar a cabo las tareas del proyecto son 2299,51 euros.

Hemos apostado por productos de contrastada marca y suficiente potencia para todas las tareas necesarias y requeridas por el administrador, que le aseguren equipo durante varios años así como facilidad de actualización en caso de necesitarlo y con buen soporte y garantía.

## 8. CRONOGRAMA.

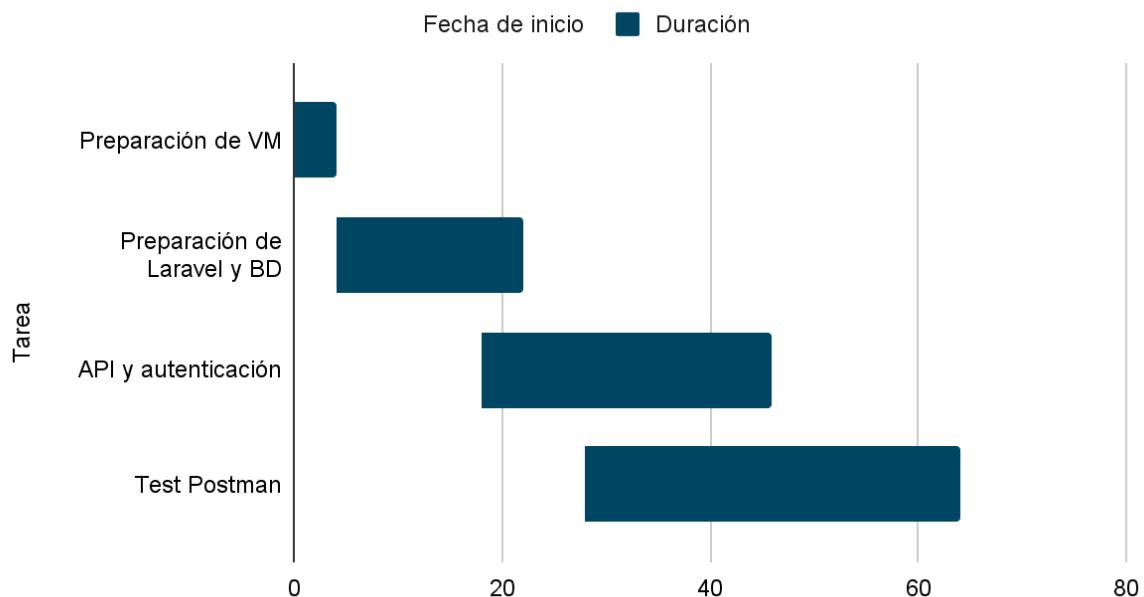
Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

## 8. Cronograma.

### Fecha de inicio y Duración



| Tarea                       | Fecha de inicio | Fecha de finalización |
|-----------------------------|-----------------|-----------------------|
| Preparación de VM           | 2/5/2022        | 6/5/2022              |
| Preparación de Laravel y BD | 13/5/2022       | 20/5/2022             |
| API y autenticación         | 23/5/2022       | 30/5/2022             |
| Test Postman                | 31/5/2022       | 8/6/2022              |

- Preparación de VM: 4 días estimados.
- Preparación de Laravel y BD: 6 días estimados.
- API y autenticación: 6 días estimados.
- Test Postman: 6 días estimados.

Todo ello sin contar la corrección de errores que puedan surgir en el proceso.

## 8. BIBLIOGRAFÍA.

Dpto. INFORMATICA / MÓDULO PROYECTO -

C.F.G.S. A.S.I.R. / Desarrollo de una API en Laravel bajo un entorno virtualizado

Andrés Felipe Rivera Riaño, Huelva, Curso 2021/2022.

## 9. Bibliografía.

Concepto de API-First:

- <https://www.bbvaapimarket.com/es/mundo-api/api-first-el-nexo-de-union-entre-la-web-y-el-movil/>
- <https://dzone.com/articles/what-api-first>

Postman:

- <https://www.postman.com/product/what-is-postman/>
- <https://lamadriguerabit.com/articulos/que-es-postman/#:~:text=%C2%BFQu%C3%A9%20es%20Postman%3F%201%20Postman.%20Se%20trata%20de,...%207%20Premier%20ejemplo.%20...%208%20Conclusiones.%20>

Información sobre hipervisores:

- <https://www.hostdime.com.ar/blog/hipervisores-tipo-1-y-tipo-2-lo-que-los-hace-diferentes/>

Contraseña segura, requisitos:

- <http://blog.banesco.com/2-formas-efectivas-de-crear-una-contraseña-segura/>

Maria-DB:

- <https://es.wikipedia.org/wiki/MariaDB>
- <https://mariadb.org/>

VMware ESXi:

- <https://www.vmware.com/products/esxi-and-esx.html>
- [https://en.wikipedia.org/wiki/VMware\\_ESXi](https://en.wikipedia.org/wiki/VMware_ESXi)

CentOS Project:

- <https://www.centos.org/>

Cómo crear una VM con VMWare ESXi:

- <https://alexariza.net/tutorial/mi-primerma-maquina-virtual/>

SELINUX:

- <https://linuxize.com/post/how-to-disable-selinux-on-centos-7/>

SSH connection refused:

- <https://serverfault.com/questions/538037/sshd-service-fails-to-start>
- <https://www.ibm.com/support/pages/sshd-start-failed-cause-start-sshdvaremptysshd-must-be-owned-root-and-not-gruop-or-world-writable>

Laravel:

- <https://laravel.com/docs/9.x>
- <https://www.edureka.co/community/65492/describe-fillable-attribute-in-a-laravel-model>
- <https://styde.net/controladores-en-laravel/>

ProFTPD:

- <https://www.redeszone.net/tutoriales/servidores/proftpd/>

Filezilla Project:

- <https://filezilla-project.org>

API Rest in Laravel 9:

- <https://www.mulesoft.com/resources/api/what-is-an-api>
- <https://www.contentful.com/blog/2021/08/12/what-is-an-api/>
- <https://www.youtube.com/watch?v=VKI9Kd2Moj8&t=1010s>

Autenticación en Laravel:

- <https://larainfo.com/blogs/laravel-8-authentication-with-laravel-ui>

Error al migrar desde el servidor:

- <https://stackoverflow.com/questions/29362394/laravel-5-error-sqlstatehy000-1045-access-denied-for-user-homesteadlocal>

Permisos en MySQL:

- <https://www.hostinger.es/tutoriales/como-crear-usuario-mysql#:~:text=Para%20crear%20una%20nueva%20cuenta%20de%20usuario%20en,elimina%20estos%20privilegios%20escribiendo%20el%20comando%3A%20FLUSH%20PRIVILEGES%3B>
- <https://www.ochobitshacenunbyte.com/2021/03/11/conceder-permisos-a-los-usuarios-de-mysql-y-mariadb#:~:text=El%20formato%20para%20asignar%20los%20permisos%20es%20el,el%20acceso%20completo%20a%20la%20totalidad%20de%20MySQL>

Subida de archivos al servidor:

- <https://www.youtube.com/watch?v=PPtcVyjFC-Q&t=2s>

Salario de un administrador de sistemas:

- <https://es.talent.com/salary?job=Administrador+De+Sistemas>

Elace de GitHub donde está alojado el proyecto:

- <https://github.com/andresfrivr/Proyecto-Integrado-ASIR>