

# **CIFP César Manrique.**

Desarrollo de aplicaciones web.

3º semipresencial.

Curso académico: 2024-2025.

Profesor: Luis Martínez Ibáñez.

Desarrollo Web en Entorno Servidor.

## **Gestionar Direcciones de Correo.**

Juan Carlos Francisco Mesa



Esta obra está licenciada bajo la Licencia Creative Commons Atribución 4.0 Internacional.  
Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/> o envíe una  
carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Índice

Actividades a realizar.....	1
Instalación de entorno de Desarrollo Web mediante stack LAMP en Kubuntu 24.04.....	3
Preguntas a responder:.....	10
¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen tu aplicación? ¿Por qué?.....	10
Si en la página de introducción de datos quieres comprobar, antes de enviar los datos, que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás?.....	11
Si en esa misma página, ahora quieres comprobar que el correo electrónico introducido no se haya introducido anteriormente y ya figure en la lista, ¿qué tecnología/lenguaje utilizarás?.....	12
¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? ¿Cómo es el o los lenguajes que se usa en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?.....	13
¿Qué parámetros debes tener en cuenta para decidirte por usar una arquitectura u otra?.....	14
Si te decides por utilizar una arquitectura AMP para la aplicación ¿qué componentes necesitas instalar en tu servidor para ejecutar la aplicación? Indica algún producto concreto para cada componente.....	15
¿Qué necesitas instalar en tu ordenador para poder desarrollar la aplicación?.....	16
Si utilizas el lenguaje PHP para programar la aplicación, ¿cuál será el tipo de datos se utilizará para manipular cada una de las direcciones de correo?.....	17

## Índice de imágenes

Imagen 1: Visualización de la salida de la ejecución de inxi para mostrar las características del equipo.....	3
Imagen 2: Instalación de Apache.....	4
Imagen 3: Instalación de Apache (cont).....	4
Imagen 4: Configuración para que se inicie desde el arranque.....	5
Imagen 5: Prueba de funcionamiento de Apache.....	5
Imagen 6: Instalación de PHP.....	6
Imagen 7: Comprobación de la instalación de php.....	6
Imagen 8: Instalación de MySQL.....	7
Imagen 9: Ejecución del Script de securización.....	7
Imagen 9: Configuración para su arranque automático al iniciar el sistema.....	7
Imagen 10: Ejecución de MySQL desde la CLI.....	8
Imagen 11: Instalación de PhpMyAdmin.....	8
Imagen 12: PhpMyAdmin en ejecución.....	9

### Actividades a realizar

Quieres programar una aplicación web para gestionar una lista de direcciones de correo. En una primera aproximación, tu aplicación se compondrá de tres páginas:

- Una página de presentación, donde explicas el cometido de la aplicación y su funcionamiento. Contiene un enlace que te lleva a la siguiente página.
- Una página de introducción de datos, donde cualquiera puede darse de alta introduciendo su nombre y su dirección de correo. Contiene un formulario que, una vez rellenado, envía los datos y te lleva a la siguiente página.
- Una página de visualización de datos, en la que se muestra la lista, nombres y direcciones de correo, de todos los que se han anotado en la aplicación.

Tienes que crear un documento en el que respondas a las siguientes preguntas sobre la aplicación que vas a desarrollar:

1. ¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen tu aplicación? ¿Por qué?
2. Si en la página de introducción de datos quieres comprobar, antes de enviar los datos, que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás?
3. Si en esa misma página, ahora quieres comprobar que el correo electrónico introducido no se haya introducido anteriormente y ya figure en la lista, ¿qué tecnología/lenguaje utilizarás?
4. ¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? ¿Cómo es el o los lenguajes que se usa en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?

## Gestionar Direcciones de Correo.

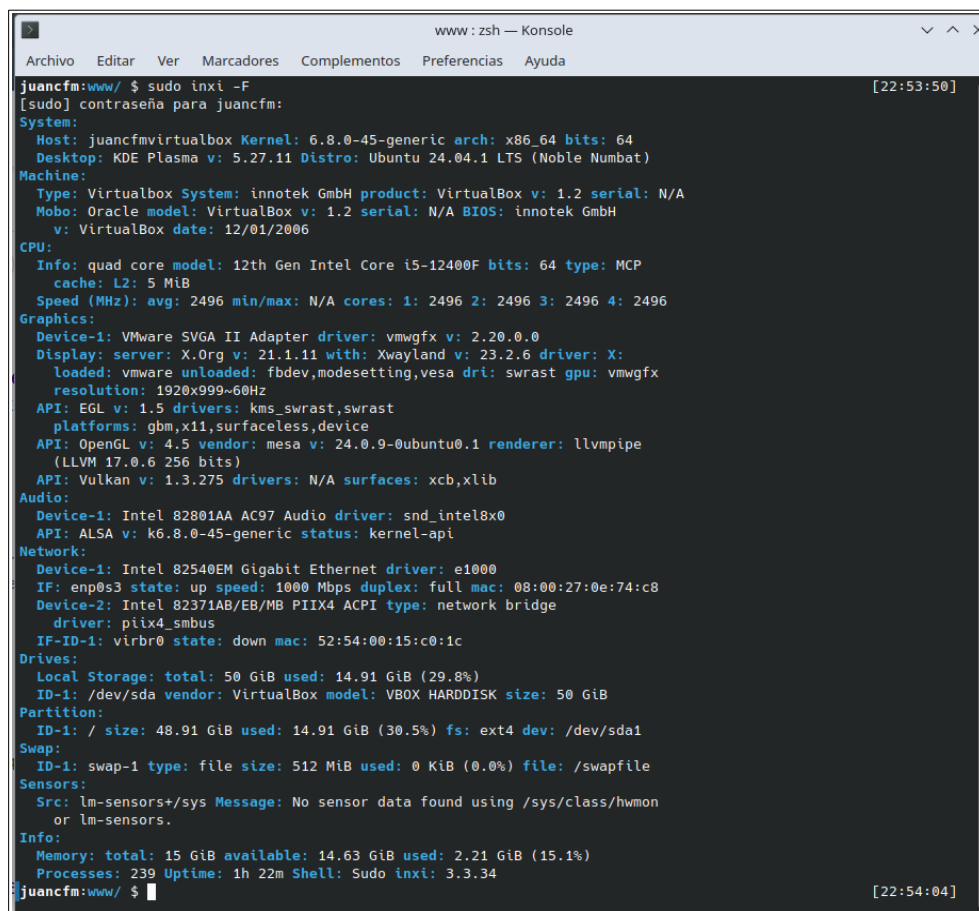
---

5. ¿Qué parámetros debes tener en cuenta para decidirte por usar una arquitectura u otra?
6. Si te decides por utilizar una arquitectura AMP para la aplicación ¿qué componentes necesitas instalar en tu servidor para ejecutar la aplicación? Indica algún producto concreto para cada componente.
7. ¿Qué necesitas instalar en tu ordenador para poder desarrollar la aplicación?
8. Si utilizas el lenguaje PHP para programar la aplicación, ¿cuál será el tipo de datos se utilizará para manipular cada una de las direcciones de correo?

[Volver al índice](#)

# Instalación de entorno de Desarrollo Web mediante stack LAMP en Kubuntu 24.04

Primero vamos a mostrar las propiedades del equipo virtual utilizado:



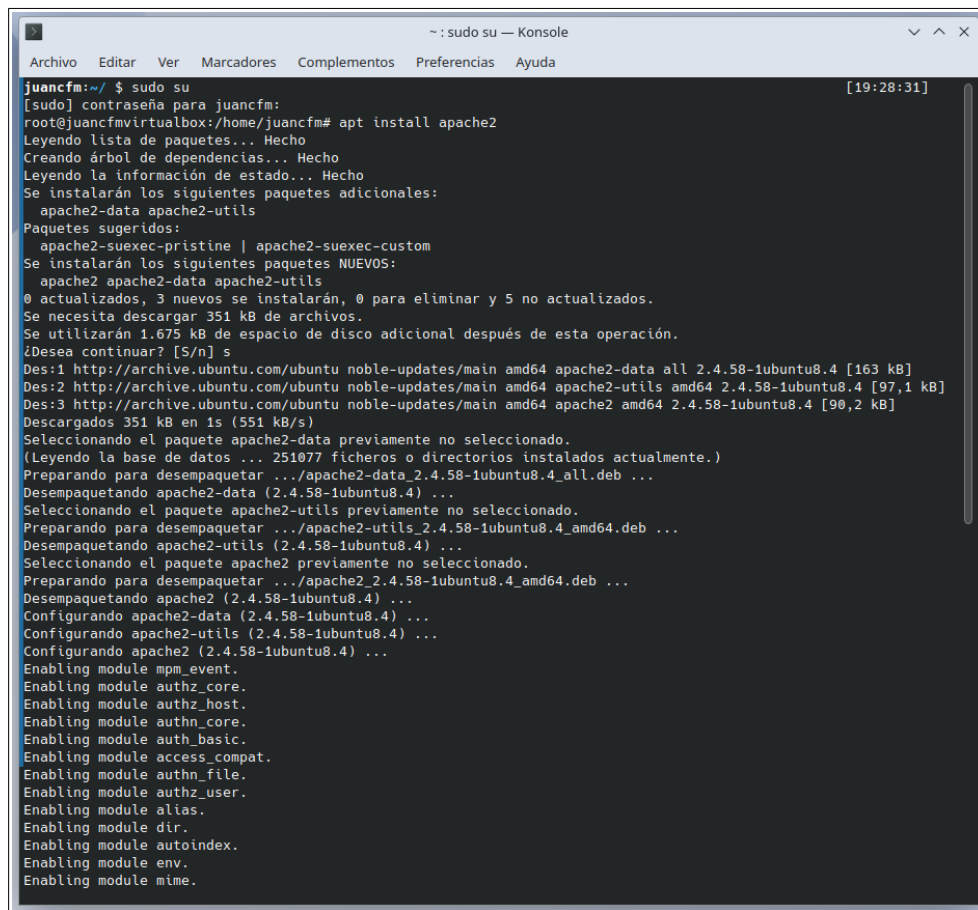
```
juancfm:www/ $ sudo inxi -F
[sudo] contraseña para juancfm:
System:
  Host: juancfmvirtualbox Kernel: 6.8.0-45-generic arch: x86_64 bits: 64
  Desktop: KDE Plasma v: 5.27.11 Distro: Ubuntu 24.04.1 LTS (Noble Numbat)
Machine:
  Type: Virtualbox System: innotek GmbH product: VirtualBox v: 1.2 serial: N/A
  Mobo: Oracle model: VirtualBox v: 1.2 serial: N/A BIOS: innotek GmbH
    v: VirtualBox date: 12/01/2006
CPU:
  Info: quad core model: 12th Gen Intel Core i5-12400F bits: 64 type: MCP
  cache: L2: 5 MiB
  Speed (MHz): avg: 2496 min/max: N/A cores: 1: 2496 2: 2496 3: 2496 4: 2496
Graphics:
  Device-1: VMware SVGA II Adapter driver: vmwgfx v: 2.20.0.0
  Display: server: X.Org v: 21.1.11 with: Xwayland v: 23.2.6 driver: X:
    loaded: vmware unloaded: fbdev,modesetting,vesa dri: swrast gpu: vmwgfx
    resolution: 1920x999-60Hz
  API: EGL v: 1.5 drivers: kms_swrast,swrast
    platforms: gbm,x11,surfaceless,device
  API: OpenGL v: 4.5 vendor: mesa v: 24.0.9-0ubuntu0.1 renderer: llvmpipe
    (LLVM 17.0.6 256 bits)
  API: Vulkan v: 1.3.275 drivers: N/A surfaces: xcb,xlib
Audio:
  Device-1: Intel 82801AA AC97 Audio driver: snd_intel8x0
  API: ALSA v: k6.8.0-45-generic status: kernel-api
Network:
  Device-1: Intel 82540EM Gigabit Ethernet driver: e1000
  IF: enp0s3 state: up speed: 1000 Mbps duplex: full mac: 08:00:27:0e:74:c8
  Device-2: Intel 82371AB/EB/MB PIIX4 ACPI type: network bridge
    driver: piix4_smbus
  IF-ID-1: virbr0 state: down mac: 52:54:00:15:c0:1c
Drives:
  Local Storage: total: 50 GiB used: 14.91 GiB (29.8%)
  ID-1: /dev/sda vendor: VirtualBox model: VBOX HARDDISK size: 50 GiB
Partition:
  ID-1: / size: 48.91 GiB used: 14.91 GiB (30.5%) fs: ext4 dev: /dev/sda1
Swap:
  ID-1: swap-1 type: file size: 512 MiB used: 0 KiB (0.0%) file: /swapfile
Sensors:
  Src: lm-sensors+sys Message: No sensor data found using /sys/class/hwmon
    or lm-sensors.
Info:
  Memory: total: 15 GiB available: 14.63 GiB used: 2.21 GiB (15.1%)
  Processes: 239 Uptime: 1h 22m Shell: Sudo inxi: 3.3.34
juancfm:www/ $
```

Imagen 1: Visualización de la salida de la ejecución de inxi para mostrar las características del equipo

[Volver al índice](#) | [Volver a índice de imágenes](#)

## Gestionar Direcciones de Correo.

Vamos a proceder a la instalación manual de cada uno de los elementos que conforman la pila LAMP.

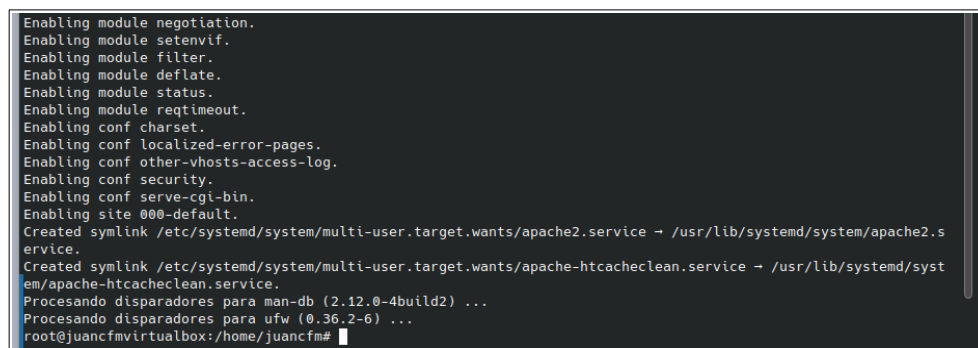


```
~: sudo su — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda

juancfm:~/$ sudo su
[sudo] contraseña para juancfm:
root@juancfmvirtualbox:/home/juancfm# apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-data apache2-utils
Paquetes sugeridos:
  apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-data apache2-utils
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
Se necesita descargar 351 kB de archivos.
Se utilizarán 1.675 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Des:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97,1 kB]
Des:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90,2 kB]
Descargados 351 kB en 1s (551 kB/s)
Seleccionando el paquete apache2-data previamente no seleccionado.
(Leyendo la base de datos ... 251077 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apache2-data_2.4.58-1ubuntu8.4_all.deb ...
Desempaquetando apache2-data (2.4.58-1ubuntu8.4) ...
Seleccionando el paquete apache2-utils previamente no seleccionado.
Preparando para desempaquetar .../apache2-utils_2.4.58-1ubuntu8.4_amd64.deb ...
Desempaquetando apache2-utils (2.4.58-1ubuntu8.4) ...
Seleccionando el paquete apache2 previamente no seleccionado.
Preparando para desempaquetar .../apache2_2.4.58-1ubuntu8.4_amd64.deb ...
Desempaquetando apache2 (2.4.58-1ubuntu8.4) ...
Configurando apache2-data (2.4.58-1ubuntu8.4) ...
Configurando apache2-utils (2.4.58-1ubuntu8.4) ...
Configurando apache2 (2.4.58-1ubuntu8.4) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
```

Imagen 2: Instalación de Apache

[Volver al índice](#) | [Volver a índice de imágenes](#)



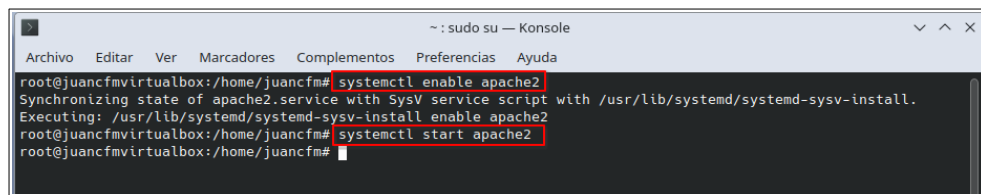
```
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.s
ervice.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/syst
em/apache-htcacheclean.service.
Procesando disparadores para man-db (2.12.0-4build2) ...
Procesando disparadores para ufw (0.36.2-6) ...
root@juancfmvirtualbox:/home/juancfm#
```

Imagen 3: Instalación de Apache (cont)

[Volver al índice](#) | [Volver a índice de imágenes](#)



# Gestionar Direcciones de Correo.



```
~ : sudo su — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda
root@juancfmvirtualbox:/home/juancfm# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
root@juancfmvirtualbox:/home/juancfm# systemctl start apache2
root@juancfmvirtualbox:/home/juancfm#
```

Imagen 4: Configuración para que se inicie desde el arranque

[Volver al índice](#) | [Volver a índice de imágenes](#)

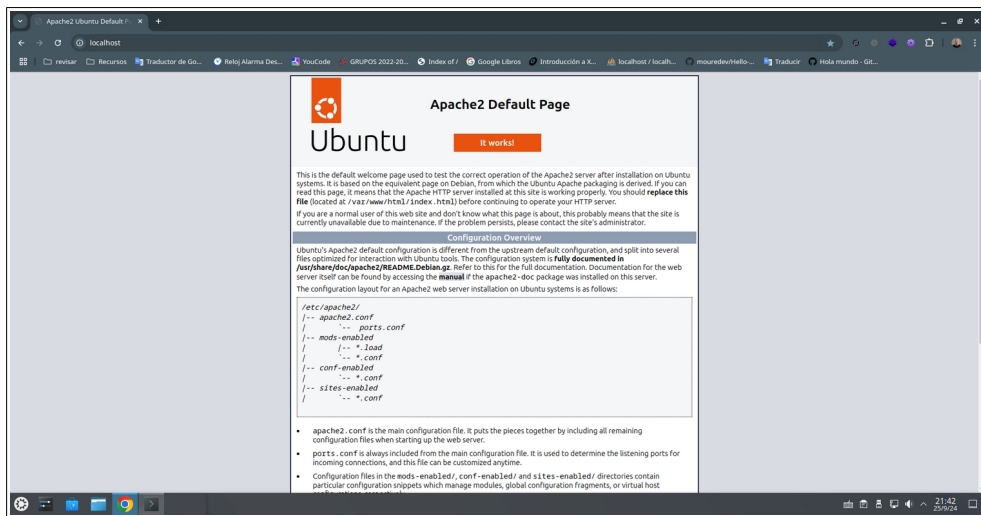
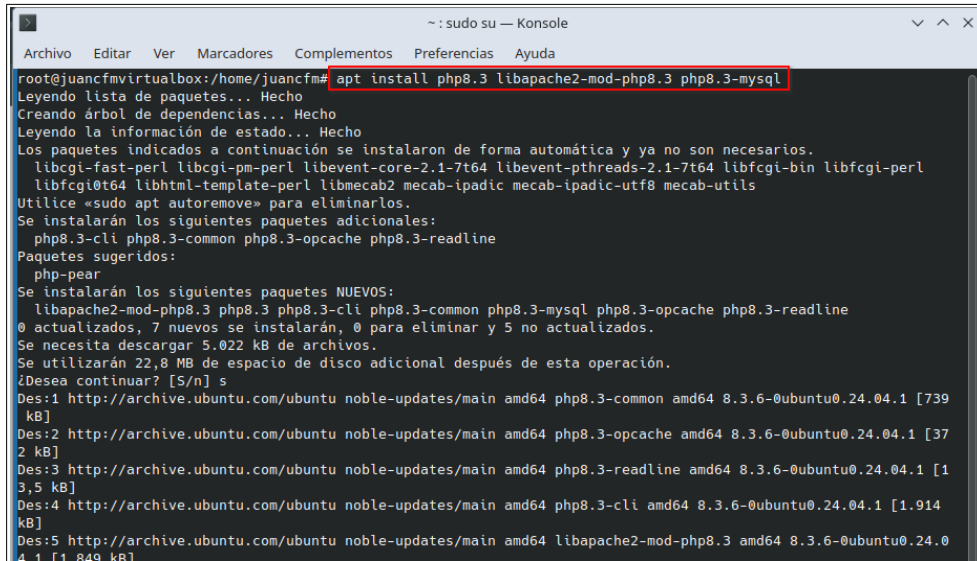


Imagen 5: Prueba de funcionamiento de Apache

[Volver al índice](#) | [Volver a índice de imágenes](#)

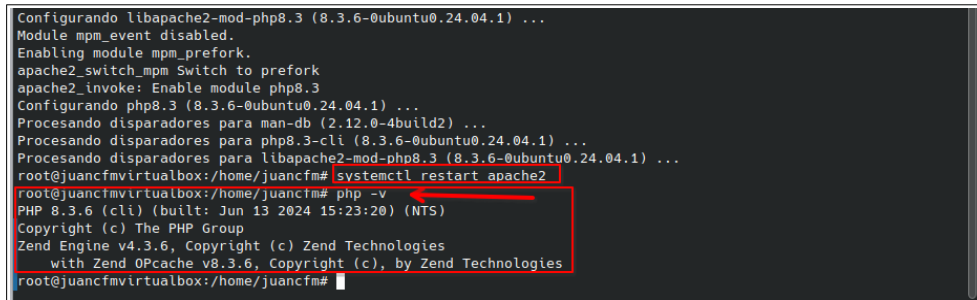
## Gestionar Direcciones de Correo.



```
~: sudo su — Konsole
Archivo Editar Ver Marcadores Complementos Preferencias Ayuda
root@juancfvirtualbox:/home/juancf# apt install php8.3 libapache2-mod-php8.3 php8.3-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7t64 libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl
  libfcgi0t64 libhtml-template-perl libmecab2 mecab-ipadic mecab-ipadic-utf8 mecab-utils
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  php8.3-cli php8.3-common php8.3-opcache php8.3-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php8.3 php8.3 php8.3-cli php8.3-common php8.3-mysql php8.3-opcache php8.3-readline
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
Se necesita descargar 5.022 kB de archivos.
Se utilizarán 22,8 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-common amd64 8.3.6-0ubuntu0.24.04.1 [739
kB]
Des:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-opcache amd64 8.3.6-0ubuntu0.24.04.1 [37
2 kB]
Des:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-readline amd64 8.3.6-0ubuntu0.24.04.1 [1
3,5 kB]
Des:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-cli amd64 8.3.6-0ubuntu0.24.04.1 [1.914
kB]
Des:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libapache2-mod-php8.3 amd64 8.3.6-0ubuntu0.24.0
4.1 [1.849 kB]
```

Imagen 6: Instalación de PHP

[Volver al índice](#) | [Volver a índice de imágenes](#)

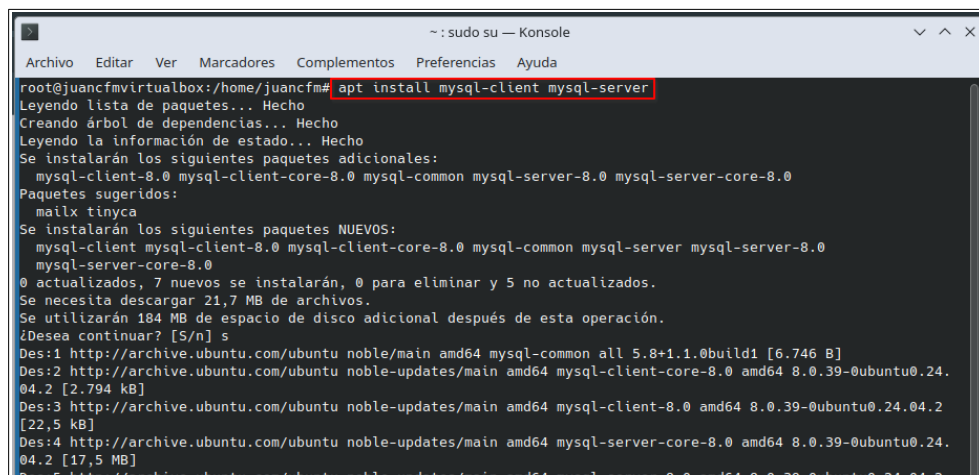


```
Configurando libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.1) ...
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.3
Configurando php8.3 (8.3.6-0ubuntu0.24.04.1) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
Procesando disparadores para php8.3-cli (8.3.6-0ubuntu0.24.04.1) ...
Procesando disparadores para libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.1) ...
root@juancfvirtualbox:/home/juancf# systemctl restart apache2
root@juancfvirtualbox:/home/juancf# php -v
PHP 8.3.6 (cli) (built: Jun 13 2024 15:23:20) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
root@juancfvirtualbox:/home/juancf#
```

Imagen 7: Comprobación de la instalación de php

[Volver al índice](#) | [Volver a índice de imágenes](#)

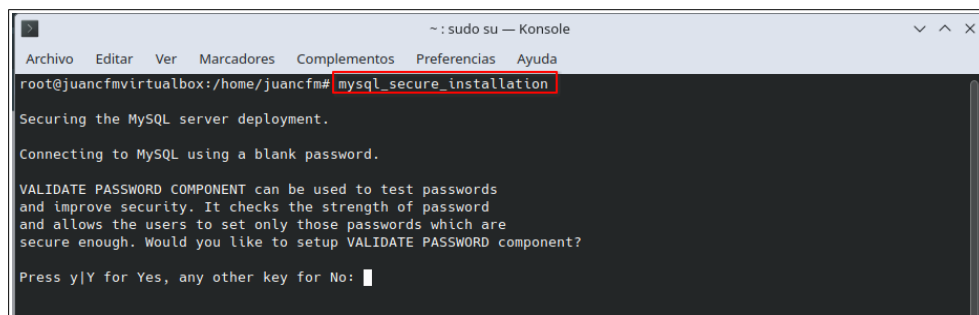
## Gestionar Direcciones de Correo.



```
~: sudo su — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda
root@juancfvirtualbox:/home/juancf# apt install mysql-client mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Paquetes sugeridos:
  mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
  mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
Se necesita descargar 21,7 MB de archivos.
Se utilizarán 184 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1 [6.746 B]
Des:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.24.
04.2 [2.794 kB]
Des:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.39-0ubuntu0.24.04.2
[22,5 kB]
Des:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server-core-8.0 amd64 8.0.39-0ubuntu0.24.
04.2 [17,5 MB]
Des:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-server-8.0 amd64 8.0.39-0ubuntu0.24.04.2
[17,5 MB]
```

Imagen 8: Instalación de MySQL

[Volver al índice](#) | [Volver a índice de imágenes](#)



```
~: sudo su — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda
root@juancfvirtualbox:/home/juancf# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

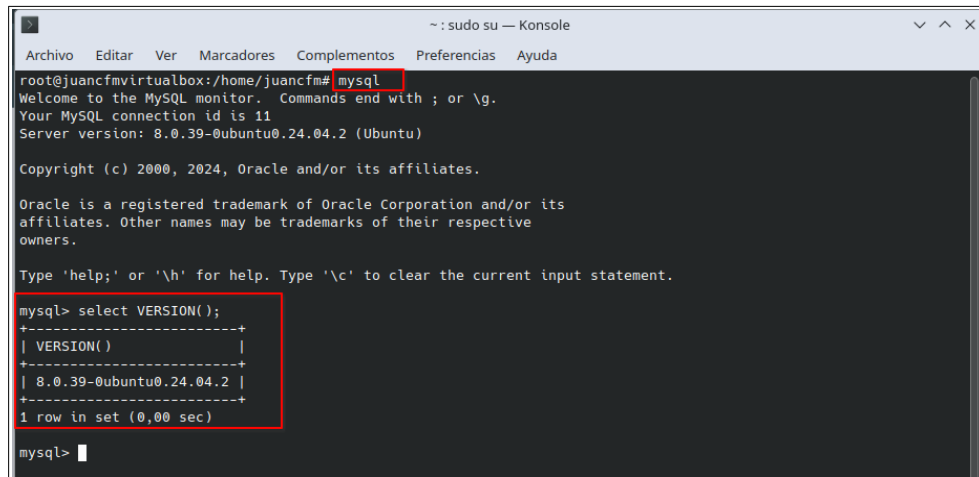
VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: 
```

Imagen 9: Ejecución del Script de securización

[Volver al índice](#) | [Volver a índice de imágenes](#)

## Gestionar Direcciones de Correo.



```
~: sudo su — Konsole
Archivo Editar Ver Marcadores Complementos Preferencias Ayuda
root@juancfvirtualbox:/home/juancf# mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.39-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

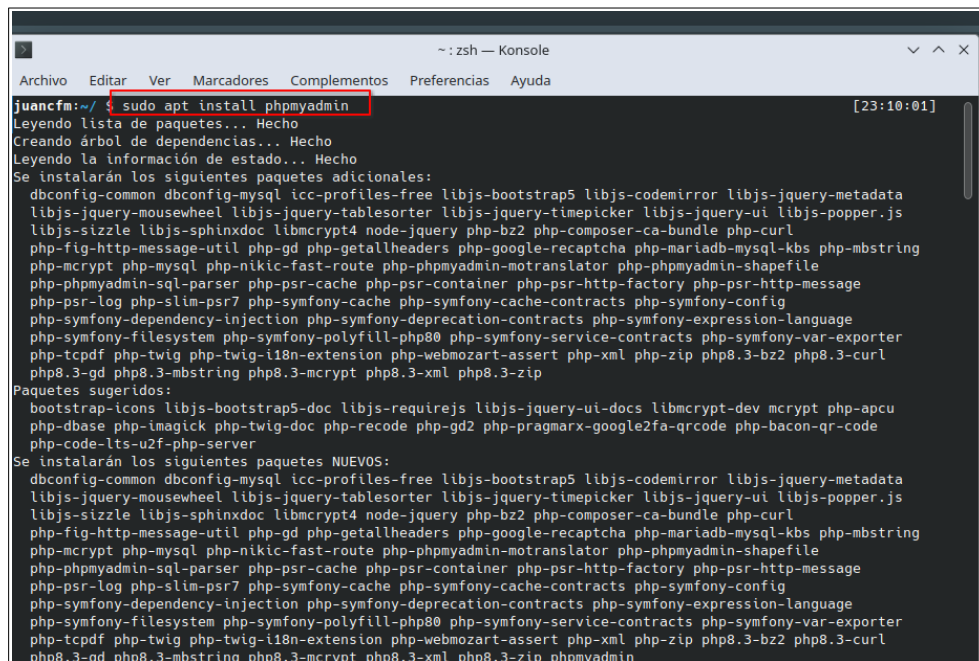
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select VERSION();
+-----+
| VERSION() |
+-----+
| 8.0.39-0ubuntu0.24.04.2 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Imagen 10: Ejecución de MySQL desde la CLI

[Volver al índice](#) | [Volver a índice de imágenes](#)



```
~: zsh — Konsole
Archivo Editar Ver Marcadores Complementos Preferencias Ayuda
juancf@juancfvirtualbox:~$ sudo apt install phpmyadmin [23:10:01]
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
dbconfig-common dbconfig-mysql icc-profiles-free libjs-bootstrap5 libjs-codemirror libjs-jquery-metadata
libjs-jquery-mousewheel libjs-jquery-tablesorter libjs-jquery-timemachine libjs-jquery-ui libjs-popover.js
libjs-sizzle libjs-sphinxdoc libmccrypt4 node-jquery php-bz2 php-composer-ca-bundle php-curl
php-fig-http-message-util php-gd php-getallheaders php-google-recaptcha php-mariadb-mysql-kbs php-mbstring
php-mcrypt php-mysql php-nikic-fast-route php-phpmyadmin-motranslator php-phpmyadmin-shapefile
php-phpmyadmin-sql-parser php-psr-cache php-psr-container php-psr-http-factory php-psr-http-message
php-psr-log php-slim-psr7 php-symfony-cache php-symfony-cache-contracts php-symfony-config
php-symfony-dependency-injection php-symfony-deprecation-contracts php-symfony-expression-language
php-symfony-filesystem php-symfony-polyfill-php80 php-symfony-service-contracts php-symfony-var-exporter
php-tcpdf php-twig php-twig-i18n-extension php-webmozart-assert php-xml php-zip php8.3-bz2 php8.3-curl
php8.3-gd php8.3-mbstring php8.3-mcrypt php8.3-xml php8.3-zip
Paquetes sugeridos:
bootstrap-icons libjs-bootstrap5-doc libjs-requirejs libjs-jquery-ui-docs libmccrypt-dev mcrypt php-apcu
php-dbase php-imagick php-twig-doc php-recode php-gd2 php-pragmarx-google2fa-qrcode php-bacon-qr-code
php-code-lts-u2f-php-server
Se instalarán los siguientes paquetes NUEVOS:
dbconfig-common dbconfig-mysql icc-profiles-free libjs-bootstrap5 libjs-codemirror libjs-jquery-metadata
libjs-jquery-mousewheel libjs-jquery-tablesorter libjs-jquery-timemachine libjs-jquery-ui libjs-popover.js
libjs-sizzle libjs-sphinxdoc libmccrypt4 node-jquery php-bz2 php-composer-ca-bundle php-curl
php-fig-http-message-util php-gd php-getallheaders php-google-recaptcha php-mariadb-mysql-kbs php-mbstring
php-mcrypt php-mysql php-nikic-fast-route php-phpmyadmin-motranslator php-phpmyadmin-shapefile
php-phpmyadmin-sql-parser php-psr-cache php-psr-container php-psr-http-factory php-psr-http-message
php-psr-log php-slim-psr7 php-symfony-cache php-symfony-cache-contracts php-symfony-config
php-symfony-dependency-injection php-symfony-deprecation-contracts php-symfony-expression-language
php-symfony-filesystem php-symfony-polyfill-php80 php-symfony-service-contracts php-symfony-var-exporter
php-tcpdf php-twig php-twig-i18n-extension php-webmozart-assert php-xml php-zip php8.3-bz2 php8.3-curl
php8.3-gd php8.3-mbstring php8.3-mcrypt php8.3-xml php8.3-zip phpmyadmin
```

Imagen 11: Instalación de PhpMyAdmin

[Volver al índice](#) | [Volver a índice de imágenes](#)

# Gestionar Direcciones de Correo.

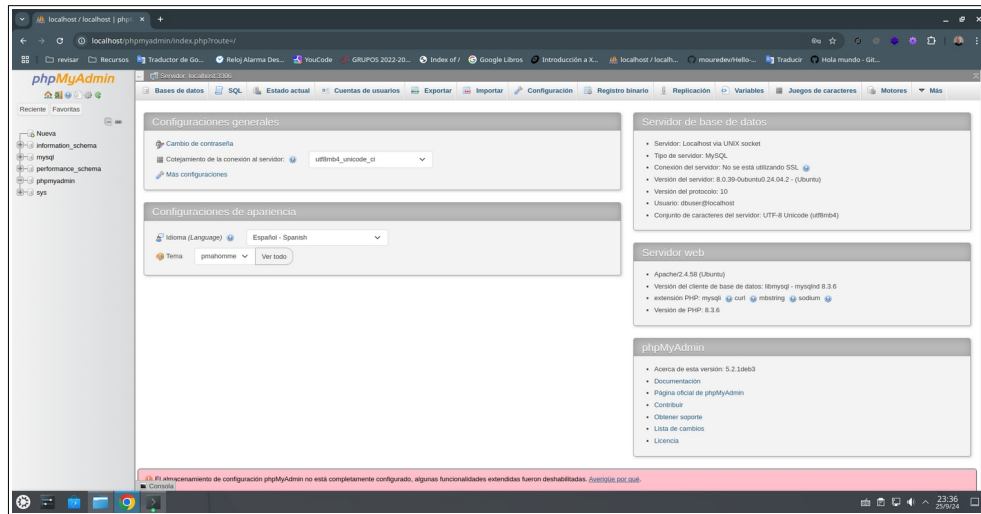


Imagen 12: PhpMyAdmin en ejecución

[Volver al índice](#) | [Volver a índice de imágenes](#)

## Preguntas a responder:

**¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen tu aplicación? ¿Por qué?**

### **Página de presentación: (index.php)**

***Estática.*** Esta página solo proporciona una explicación del funcionamiento de la aplicación. El contenido es fijo y no cambia según el usuario.

### **Página de introducción de datos: (register.php)**

***Dinámica.*** Aquí se requiere un formulario donde los usuarios introducen su nombre y dirección de correo, y se debe procesar y validar esa información, lo que implica interacción con el servidor. Los datos se almacenan, por lo que es una página dinámica.

### **Página de visualización de datos: (list.php)**

***Dinámica.*** Esta página muestra una lista de direcciones de correo que puede cambiar continuamente a medida que más usuarios se registran, por lo que requiere extraer la información desde una base de datos o una fuente similar. Por lo tanto, el contenido es dinámico.

[Volver al índice](#)

**Si en la página de introducción de datos quieres comprobar, antes de enviar los datos, que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás?**

Utilizaría JavaScript para realizar la validación del correo en el lado del cliente. Esto permite una validación instantánea antes de enviar los datos al servidor, mejorando la experiencia del usuario. JavaScript puede usarse para asegurarse de que el correo electrónico cumple con ciertos requisitos, como la presencia de un símbolo "@", usando expresiones regulares.

También utilizaría HTML5, que proporciona etiquetas y atributos para realizar validaciones simples sin la necesidad de JavaScript. Para validar que el correo electrónico tiene una estructura correcta, se puede utilizar el atributo `type="email"` dentro de la etiqueta `<input>`. Este tipo de campo se asegura de que la entrada tenga el formato adecuado de un correo electrónico (por ejemplo, que contenga el símbolo "@" y un dominio).

[Volver al índice](#)

**Si en esa misma página, ahora quieres comprobar que el correo electrónico introducido no se haya introducido anteriormente y ya figure en la lista, ¿qué tecnología/lenguaje utilizarás?**

Para comprobar que el correo no esté duplicado, es necesario verificar contra la base de datos, lo que requiere una tecnología del lado del servidor como PHP. Una vez que el usuario envíe el formulario, el servidor puede consultar la base de datos para ver si el correo ya existe y devolver una respuesta antes de continuar con el registro.

[Volver al índice](#)



**¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? ¿Cómo es el o los lenguajes que se usa en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?**

**LAMP (Linux, Apache, MySQL, PHP):** PHP es un lenguaje de guiones (interpretado) que se ejecuta en el servidor.

**ASP.NET:** Usa C# o VB.NET, que son lenguajes compilados a código intermedio (MSIL) y ejecutados por el Common Language Runtime (CLR).

**Java (JSP/Servlets):** Java se compila a un código intermedio (bytecode) que es ejecutado por la Máquina Virtual de Java (JVM).

[Volver al índice](#)

## ¿Qué parámetros debes tener en cuenta para decidirte por usar una arquitectura u otra?

**Costo:** Algunas arquitecturas como LAMP (PHP) o MEAN (Node.js) son de código abierto y gratuitas, mientras que otras como ASP.NET pueden requerir licencias.

**Rendimiento:** Si se necesita manejar una alta concurrencia o un sistema que escala fácilmente, arquitecturas como MEAN (con Node.js) o Java podrían ser más adecuadas.

**Facilidad de desarrollo:** Arquitecturas como LAMP son populares por su facilidad de uso y la abundancia de recursos, mientras que Java y .NET ofrecen características robustas para aplicaciones a gran escala.

**Compatibilidad:** Si la aplicación necesita correr en un entorno específico (por ejemplo, servidores Windows), se puede optar por tecnologías como ASP.NET.

**Escalabilidad:** Si se espera un alto volumen de tráfico, una arquitectura basada en Node.js o Java podría ser más adecuada debido a su capacidad para manejar muchas conexiones simultáneas.

[Volver al índice](#)

**Si te decides por utilizar una arquitectura AMP para la aplicación ¿qué componentes necesitas instalar en tu servidor para ejecutar la aplicación? Indica algún producto concreto para cada componente.**

La arquitectura AMP (Apache, MySQL, PHP) requiere los siguientes componentes:

**Servidor web: Apache** HTTP Server. Es el servidor web que manejará las solicitudes HTTP.

**Base de datos: MySQL o MariaDB** para gestionar la información que los usuarios introducen, como los correos electrónicos.

**Lenguaje de programación del servidor: PHP**, que se encargará de la lógica del servidor, como el procesamiento de formularios y la comunicación con la base de datos.

[Volver al índice](#)

## ¿Qué necesitas instalar en tu ordenador para poder desarrollar la aplicación?

**Entorno de desarrollo local:** Instalar un servidor AMP local como XAMPP o MAMP que incluya Apache, MySQL y PHP.

**Editor de código:** Un editor o entorno de desarrollo integrado (IDE) como VS Code, Sublime Text o PHPStorm.

**Navegador:** Para probar la aplicación.

**Gestor de bases de datos:** Se puede usar una herramienta como **phpMyAdmin** (incluida en XAMPP) para administrar la base de datos MySQL o instalar una herramienta independiente como **DBeaver** o **MySQL Workbench**.

[Volver al índice](#)

## **Si utilizas el lenguaje PHP para programar la aplicación, ¿cuál será el tipo de datos se utilizará para manipular cada una de las direcciones de correo?**

En PHP, los correos electrónicos se manejarán como cadenas de texto (strings). Un correo electrónico es esencialmente una secuencia de caracteres alfanuméricos, y PHP proporciona funciones para trabajar con cadenas, lo que es útil para validar o manipular direcciones de correo.

Los datos se almacenarán en un archivo JSON y serán tratados como una estructura de tipo array asociativo cuando se decodifiquen.

[Volver al índice](#)