

# Ejercicio 1 (RA1.a, RA1.b, RA1.d).

Voy a elegir una página como ejemplo para analizar y responder las preguntas. La página seleccionada es [Renfe](#), que es la empresa de transporte ferroviario en España.

---

**¿De qué negocio o empresa se trata? (indica la URL)**

La empresa es [Renfe](#).

---

**¿Qué se puede hacer en su página web?**

1. Comprar billetes de tren.
  2. Consultar horarios y precios.
  3. Comprar y cambiar de billetes.
- 

**¿Qué aspectos negativos tiene su página web?**

1. **Velocidad de carga lenta:** Sobre todo cuando tiene que hacer consultas de disponibilidad
  2. **Errores en las búsquedas:** Cuando buscas billetes, a veces el sistema da errores o es demasiado lento.
- 

**¿Tienen contenido dinámico (generación dinámica) o solo estático?**

La página tiene contenido dinámico, especialmente en las búsquedas de billetes y los formularios de consulta.

---

**Indica al menos dos mejoras que realizarías en dicha página que impliquen la ejecución de código en el servidor.**

1. **Optimizar el sistema de búsqueda de billetes:** Reducir los tiempos de respuesta implementando paginación o cacheando con cron.

2. **Gestionar mejor los errores:** Implementar mensajes de error más detallados y y user friendly.
- 

**Indica cómo podría ayudarte un Framework en la implementación de las mejoras anteriores.**

1. **Optimización de búsqueda de billetes:**
  - Usando un framework como Laravel, se puede implementar almacenamiento en caché (por ejemplo Redis) para guardar resultados frecuentes y reducir la carga del servidor.
  - Realizar un cron job en intervalos basados en las consultas que reciben cada X y almacene en cache la disponibilidad.
2. **Manejo de errores:**
  - Laravel tiene un sistema integrado para la gestion de errores y facilita la configuración de API RESTful para enviar respuestas como códigos HTTP y detalles del error.

## Ejercicio 2 (RA1.a, RA1.c).

Fíjate en el siguiente ejemplo que describe la interacción entre navegador (cliente web) y servidor web donde se detalla lo que ocurre en cada una de las partes. Intenta pensar en otro "Caso de uso" adaptado a la web elegida por ti en el ejercicio anterior donde un potencial usuario (o miembro de la organización) utilice la aplicación web en un contexto de generación dinámica de contenido.

### ***Caso de uso: Un voluntario rellena el formulario de contacto***

1. La persona está rellenando el formulario de registro en una página web y ha terminado de rellenarlo con su email y contraseña para registrarse.
2. El navegador realiza una petición tipo HTTP POST al servidor web.
3. El servidor web recibe la petición con los datos del formulario, detecta que esa petición tiene que pasar por un script PHP que valide los datos.
4. El motor PHP se arranca e inicia la ejecución del script el cual contiene:
  - Código para comprobar si el email ya está registrado en la base de datos.
  - Código para comprobar si la contraseña cumple con los requisitos de seguridad.

- Código para hashear la contraseña.
  - Código para almacenar los datos (email y contraseña hasheada) en la base de datos.
  - Código para generar HTML con la respuesta al usuario, indicando si el registro fue exitoso o si hubo un error.
5. Una vez que el motor PHP ha terminado de ejecutar el script, el HTML generado es enviado al servidor.
  6. El servidor web recoge el resultado generado por el motor PHP y lo envía al navegador/cliente con una respuesta HTTP.
  7. El navegador HTTP recibe la respuesta que incluye el HTML y lo renderiza para que la persona vea el mensaje de confirmación o error.

## Ejercicio 3 (RA1.e, RA1.g, RA2.b).

Rellena la siguiente tabla indicando **casos concretos** en los que podría ser necesario emplear las siguientes tecnologías en la mejora de la web elegida en el ejercicio 1:

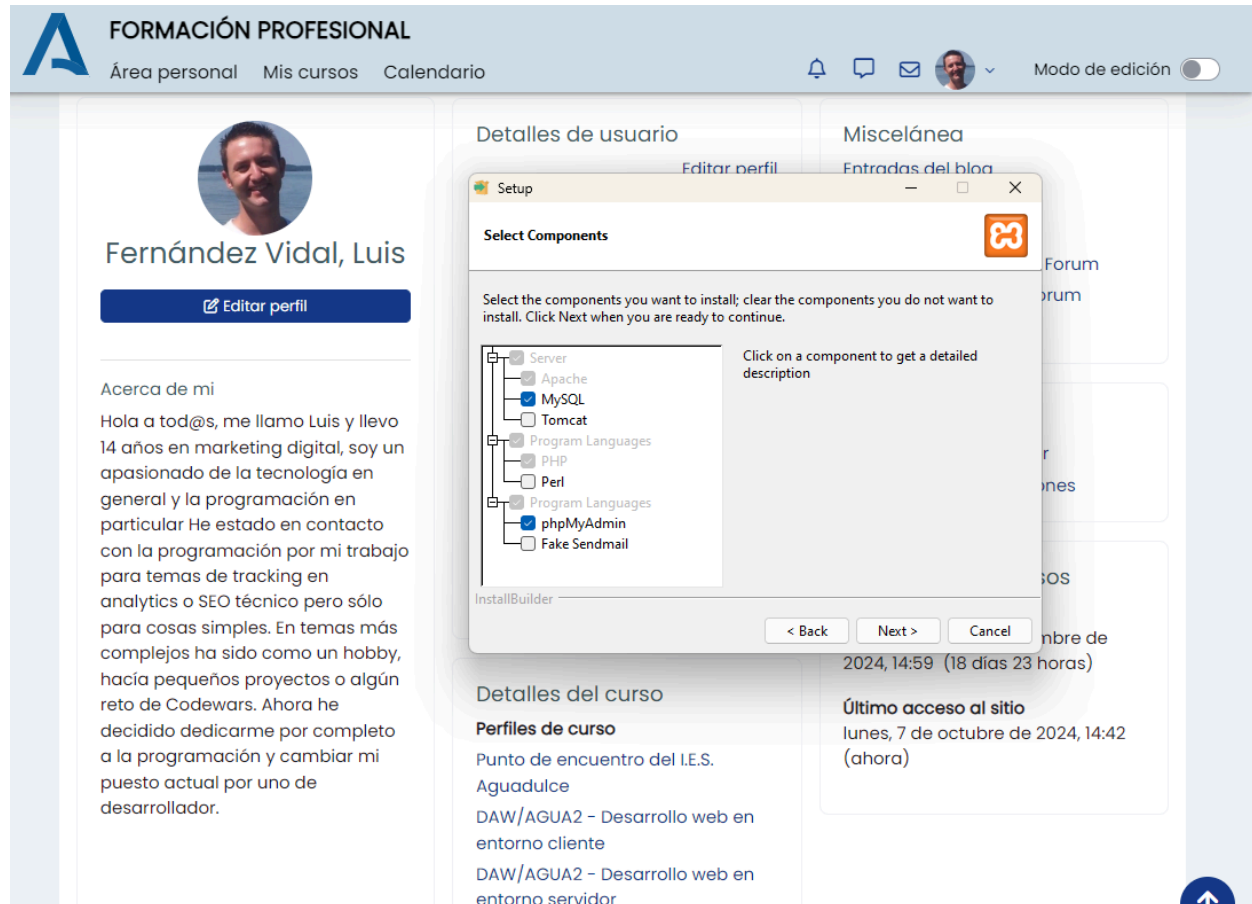
Tecnología	¿Dónde y como se usaría?
HTTP	HTTP podría usarse en las páginas generales de Renfe que no gestionen datos sensibles como mostrar horarios
HTTPS	HTTPS se usaría en todo el proceso de compra de billetes, incluyendo dar datos personales y datos para el pago o registrarse como usuario
HTML	HTML se usaría para estructurar las páginas de la web de Renfe, como la página de inicio, la de búsqueda de billetes, los formularios de compra
PHP	PHP se usaría en el procesamiento del formulario de búsqueda y compra de billetes. Además, gestionaría la validación de los datos (como fechas de viaje o la disponibilidad de trenes)

CSS	CSS se usaría para diseñar la interfaz dándole estilo, haciendola responsive y user friendly
SQL	SQL se usaría para consultar y actualizar la base de datos de Renfe.
JavaScript	Se usaría para mejorar la interacción del usuario en la web de Renfe, como la validación en tiempo real de los formularios de reserva, la actualización dinámica de los horarios y precios sin necesidad de recargar la página (AJAX)

## Ejercicio 4 (RA1.f, RA2.a, RA2.b).

Realiza la instalación de [XAMPP 8.2.4](#) en tu ordenador (preferiblemente en **C:/XAMPP**) y realiza lo siguiente:

- Realiza una captura de pantalla durante el proceso de instalación.



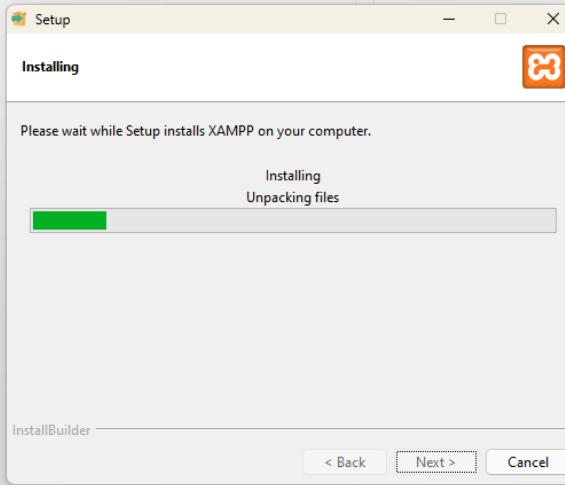


Fernández Vidal, Luis

[Editar perfil](#)

## Acerca de mi

Hola a tod@s, me llamo Luis y llevo 14 años en marketing digital, soy un apasionado de la tecnología en general y la programación en particular. He estado en contacto con la programación por mi trabajo para temas de tracking en analytics o SEO técnico pero sólo para cosas simples. En temas más complejos ha sido como un hobby, hacía pequeños proyectos o algún reto de Codewars. Ahora he decidido dedicarme por completo a la programación y cambiar mi puesto actual por uno de desarrollador.



## Detalles del curso

### Perfiles de curso

Punto de encuentro del I.E.S.

Aguadulce

DAW/AGUA2 - Desarrollo web en entorno cliente

DAW/AGUA2 - Desarrollo web en entorno servidor

DAW/AGUA2 - Despliegue de

### Primer acceso al sitio

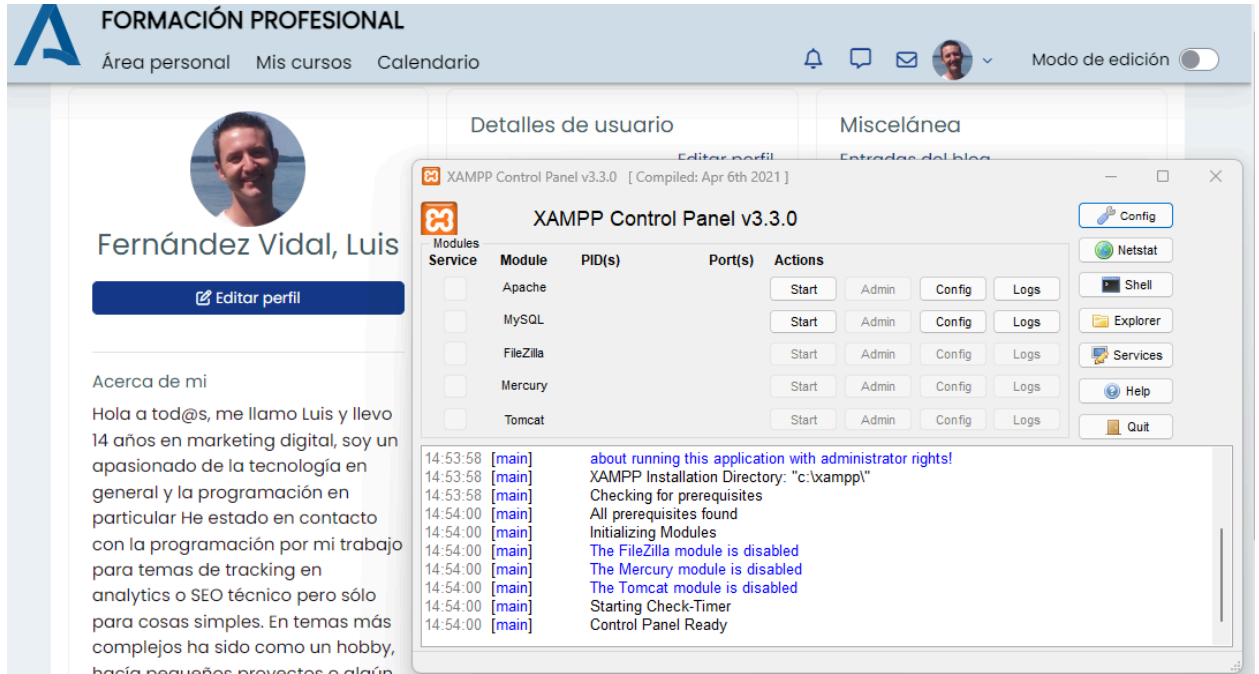
miércoles, 18 de septiembre de 2024, 14:59 (18 días 23 horas)


### Último acceso al sitio

lunes, 7 de octubre de 2024, 14:42 (ahora)



- Realiza una captura de pantalla de la interfaz de XAMPP para arrancar o detener el servidor web.




- 

FORMACIÓN PHP 8.2

Área personal

Mis cursos

Calendario



Fernández Vidal, Luis

✎ Editar perfil

Acerca de mí

Hola a todos, me llamo Luis y llevo 14 años en marketing digital, soy un apasionado de la tecnología en general y la programación en particular. He estado en contacto con la programación por mi trabajo para temas de tracking en analytics o SEO técnico pero sólo para cosas simples. En temas más complejos he sido como un hobby, hacía pequeños proyectos o algún reto de Codewars. Ahora he decidido dedicarme por completo a la programación y cambiar mi puesto actual por uno de desarrollador.

Información Personal

**Dirección de correo:**  
fernandezvidalluis@gmail.com

**País:**  
España

**Ciudad:**  
Sevilla

PHP 8.2.4 - phpinfo.php

localhost/phpinfo.php

Next Courses

Surf

DAW

Code

Tools

SEO


Roadmap

RoadM

Type

All Bookmarks

PHP Version 8.2.4



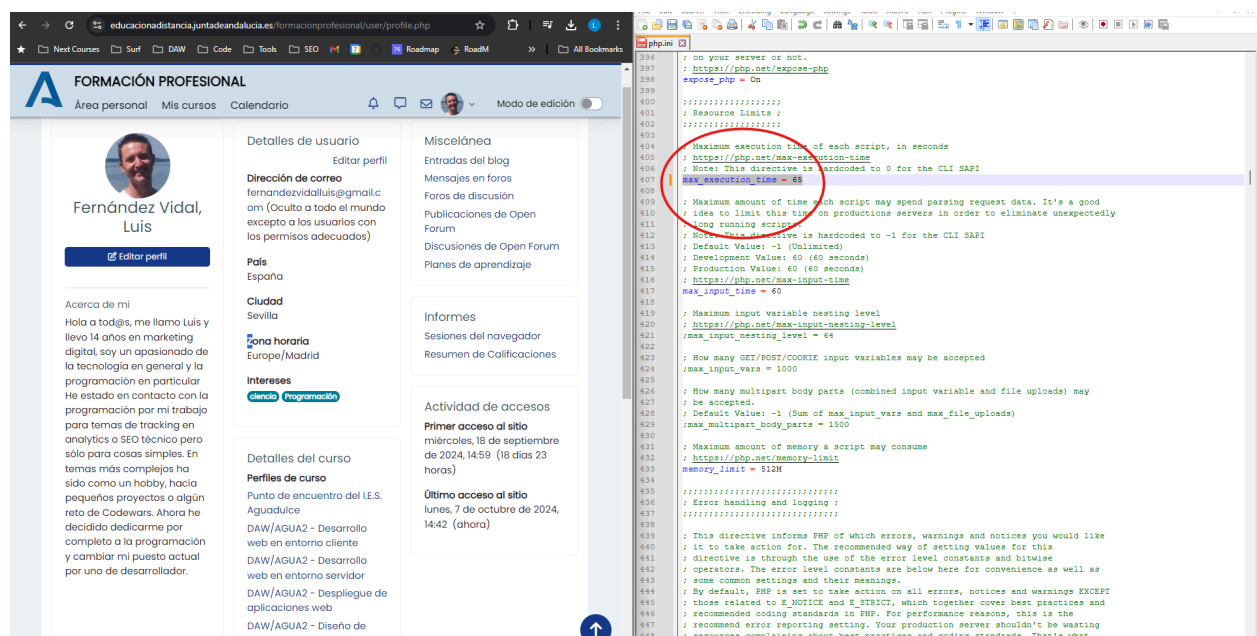
System	Windows NT LAPT0P-B1D4JH0 10.0 build 22H2 (Windows 11) AMD64
Build Date	Mar 14 2023 17:50:26
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	script_hologo /e:script configure.js "--enable-snapshot-build"--enable-debug-pack"--with-pdo-odbc=/L/L/L/instanclentfcdk_shared"--with-oci8-19=/L/L/L/instanclentfcdk_shared"--enable-object-out-dir="--enable-com-dtnet-shared"--without-analyzer"--with-pgsql
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829.TS.VS16
PHP Extension Build	API20220829.TS.VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*



## Ejercicio 5 (RA1.d, RA2.f).

Localiza el archivo `php.ini` (generalmente en `c:/xampp/php/php.ini`). Este archivo contiene un montón de directivas que determinan como trabaja el motor PHP cuando es invocado por el servidor Apache. La configuración de PHP incluida por defecto viene establecida para un escenario de desarrollo (cuando la aplicación pasa a producción se pone una configuración más segura y fiable). El objetivo ahora es modificar dos directivas que afectarán al funcionamiento de PHP:

- Modifica la directiva `max_execution_time` que determina el tiempo máximo que un script PHP puede ejecutarse, cambiando el valor a 65 segundos. Realiza una captura de pantalla de la modificación.



- Modifica la directiva modificar los errores reportados por el usuario de forma que se quede así: `error_reporting=E_ALL & ~E_STRICT`.

The image shows a web browser window with a user profile page on the left and a code editor on the right. The profile page is for 'Fernández Vidal, Luis' and includes sections for 'Detalles de usuario', 'Miscelánea', 'Informes', and 'Actividad de accesos'. The code editor shows the PHP configuration file, with a red circle highlighting the line `error_reporting = E_ALL & ~E_STRICT`. The code editor also shows comments explaining the error reporting settings.

**Detalles de usuario**

**Dirección de correo**  
fernandezvidalluis@gmail.com (Oculto a todo el mundo excepto a los usuarios con los permisos adecuados)

**País**  
España

**Ciudad**  
Sevilla

**Zona horaria**  
Europe/Madrid

**Intereses**  
desarrollo programación

**Detalles del curso**

**Perfiles de curso**

Punto de encuentro del I.E.S. Aguadulce  
DAW/AQUA2 - Desarrollo web en entorno cliente  
DAW/AQUA2 - Desarrollo web en entorno servidor  
DAW/AQUA2 - Despliegue de aplicaciones web  
DAW/AQUA2 - Diseño de interfaces WEB

**Miscelánea**

Entradas del blog  
Mensajes en foros  
Foros de discusión  
Publicaciones de Open Forum  
Discusiones de Open Forum  
Planes de aprendizaje

**Informes**

Sesiones del navegador  
Resumen de Calificaciones

**Actividad de accesos**

**Primer acceso al sitio**  
miércoles, 18 de septiembre de 2024, 14:59 (18 días 23 horas)

**Último acceso al sitio**  
lunes, 7 de octubre de 2024, 14:42 (ahora)

```

465 // empty string)
466 // E_STRICT
467 // - run-time notices, enable to have PHP suggest changes
468 // to your code which will ensure the best interoperability
469 // and forward compatibility of your code
470 // fatal errors that occur during PHP's initial startup
471 // warnings (non-fatal errors) that occur during PHP's
472 // initial startup
473 // E_COMPILE_ERROR
474 // - fatal compile-time errors
475 // E_COMPILE_WARNING
476 // - compile-time warnings (non-fatal errors)
477 // E_USER_ERROR
478 // - user-generated error message
479 // E_USER_WARNING
480 // - user-generated warning message
481 // E_USER_NOTICE
482 // - user-generated notice message
483 // E_DEPRECATED
484 // - warn about code that will not work in future versions
485 // of PHP
486 // E_USER_DEPRECATED
487 // - user-generated deprecation warnings
488 //
489 // Common Values:
490 // E_ALL (Show all errors, warnings and notices including coding standards.)
491 // E_ALL & ~E_NOTICE (Show all errors, except for notices)
492 // E_ALL & ~E_NOTICE & ~E_STRICT (Show all errors, except for notices and coding standards warnings.)
493 // E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR (Show only errors)
494 //
495 // Default Values:
496 // E_ALL
497 // Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
498 // https://php.net/error-reporting
499 error_reporting = E_ALL & ~E_STRICT
500 //
501 // This directive controls whether or not and where PHP will output errors,
502 // notices and warnings too. Error output is very useful during development, but
503 // it could be very dangerous in production environments. Depending on the code
504 // which is triggering the error, sensitive information could potentially leak
505 // out of your application such as database usernames and passwords or worse.
506 // For production environments, we recommend logging errors rather than
507 // sending them to STDOUT.
508 // Possible Values:
509 // Off = Do not display any errors
510 // stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
511 // On or stdout = Display errors to STDOUT
512 // Default Value: On
513 // Development Value: On
514 // Production Value: Off
515 // https://php.net/display-errors
516 display_errors = On
517 //
518 // The display of errors which occur during PHP's startup sequence are handled
519 // separately from display_errors. We strongly recommend you set this to 'off'
520 // for production servers to avoid leaking configuration details.
521 // Default Value: On
522 // Development Value: On
523 // Production Value: Off
524 // https://php.net/display-startup-errors
525 display_startup_errors = On
526 
```