



Laravel

Juliana Abril, Camila Granda, Emily Romero

Tabla de Contenido

01

Introducción

Inicios de Laravel

02

¿Qué es Laravel?

Definiciones y Características

03

Arquitectura

Arquitectura usada por Laravel

04

Patrones de Diseño

05

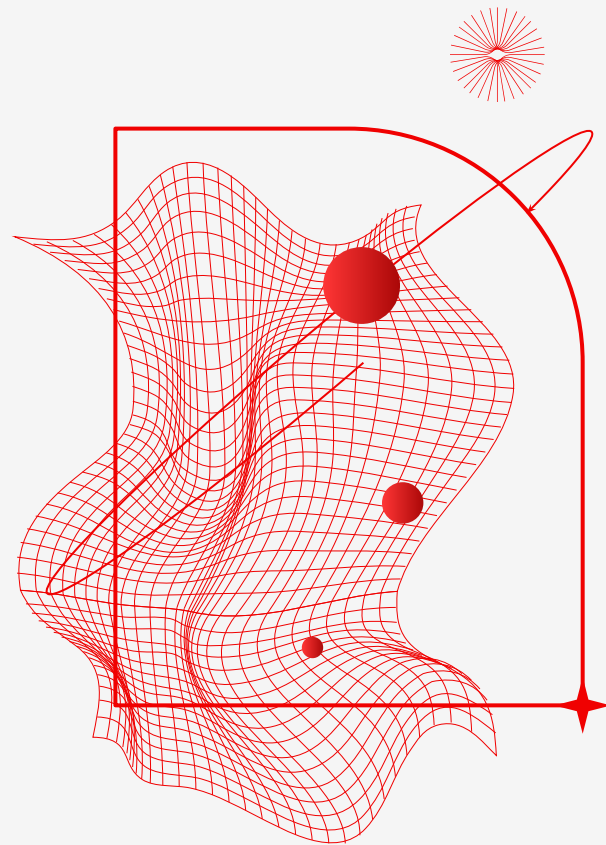
Curva de Aprendizaje



01

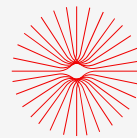
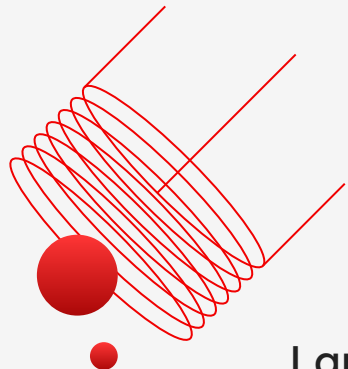
Introducción

Inicios de Laravel



Laravel

Laravel es creado por Taylor Otwell en 2011, con el objetivo de proporcionar una alternativa más avanzada y flexible a los marcos de desarrollo PHP existentes. Rápidamente se distinguió por su elegancia, sencillez y, sobre todo, por su enfoque en el desarrollador.

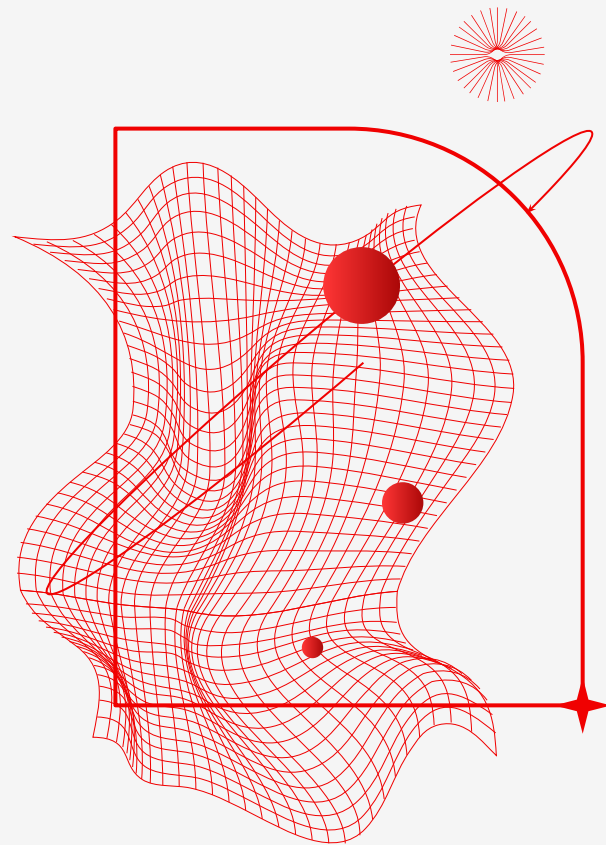




02

¿Qué es Laravel?

Definiciones y Características



¿Qué es Laravel?

Laravel es un framework de aplicaciones web con una sintaxis expresiva y elegante. Proporciona una estructura y un punto de partida para crear aplicaciones, permitiendo a los desarrolladores centrarse en crear funcionalidades sorprendentes mientras Laravel se encarga de los detalles complejos.

Laravel se esfuerza por proporcionar una experiencia de desarrollador increíble al tiempo que proporciona potentes funciones para:

- Inyecciones de dependencias.
- Capa de abstracción de base de datos expresiva.
- Manejo de colas y trabajos programados
- Herramientas para pruebas unitarias y de integración.

Inyección de Dependencias

Utiliza un patrón que permite a los desarrolladores proporcionar dependencias a una clase de manera externa, facilitando la modularidad y reutilización del código. En Laravel, esto se utiliza para gestionar las dependencias de las clases de forma automática, lo cual promueve un diseño de código más limpio y mantenible. **Por ejemplo:**

```
class ServicioDeCorreo {  
    private $clienteSMTP;  
  
    // Inyección de dependencia a través del constructor  
    public function __construct(ClienteSMTP $clienteSMTP) {  
        $this->clienteSMTP = $clienteSMTP;  
    }  
  
    public function enviarCorreo($destinatario, $asunto, $mensaje) {  
        // Utiliza el cliente SMTP inyectado para enviar el correo  
        $this->clienteSMTP->conectar();  
        $this->clienteSMTP->enviar($destinatario, $asunto, $mensaje);  
        $this->clienteSMTP->desconectar();  
    }  
}
```

Imagina que tienes una clase **ServicioDeCorreo** que necesita enviar correos electrónicos. En lugar de que esta clase cree directamente una instancia de la clase ClienteSMTP para manejar la conexión al servidor SMTP, la clase ServicioDeCorreo acepta una instancia de **ClienteSMTP** desde fuera.

Capa de abstracción de base de datos

- Laravel proporciona un ORM (Object-Relational Mapping) llamado Eloquent, que permite interactuar con la base de datos utilizando objetos y métodos en lugar de consultas SQL directas.
- Esto simplifica el manejo de datos y mejora la seguridad al prevenir ataques de SQL injection. **Por ejemplo tenemos el siguiente código**

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    protected $table = 'users'; // Nombre de la tabla en la base de datos

    protected $fillable = [
        'name', 'email', 'password',
    ];

    // Ejemplo de método en el modelo para obtener usuarios activos
    public function scopeActive($query)
    {
        return $query->where('active', true);
    }
}
```


Capa de abstracción de base de datos

- En Laravel, cada tabla de la base de datos tiene su propio modelo correspondiente en el directorio **app/Models**.
- La propiedad **'table'** especifica el nombre de la tabla en la base de datos con la que el modelo User interactúa.
- La propiedad **'fillable'** define qué campos pueden ser asignados masivamente. Solo los campos listados en fillable serán asignados cuando se use create o update.
- Eloquent permite definir **métodos de ámbito ('scopes')** que actúan como consultas predefinidas. En este caso, scopeActive define un ámbito para obtener solo usuarios activos en la base de datos.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    protected $table = 'users'; // Nombre de la tabla en la base de datos

    protected $fillable = [
        'name', 'email', 'password',
    ];

    // Ejemplo de método en el modelo para obtener usuarios activos
    public function scopeActive($query)
    {
        return $query->where('active', true);
    }
}
```

Manejo de colas y tareas programadas

- Laravel incluye un sistema robusto para manejar colas de trabajo asíncronas y tareas programadas (cron jobs).
- Esto es útil para procesar tareas en segundo plano de manera eficiente, como enviar correos electrónicos, procesar pagos, o realizar operaciones intensivas que no deben bloquear la aplicación principal.
- Laravel soporta múltiples drivers de colas como Redis, RabbitMQ, y base de datos, lo que permite adaptarse a diferentes necesidades y configuraciones de infraestructura.
 - Las tareas que se encolan se implementan como clases Job en Laravel, que encapsulan la lógica específica de la tarea a realizar.

Herramientas robustas para pruebas unitarias e integración

- Las pruebas unitarias en Laravel permiten evaluar el comportamiento individual de métodos y funciones dentro del código.
- En cambio las pruebas de integración evalúan la interacción entre diferentes componentes del sistema para garantizar que funcionen de manera conjunta como se espera.
- Laravel utiliza PHPUnit como su framework de pruebas por defecto, facilitando la escritura y ejecución de pruebas unitarias de manera estructurada y eficiente.
- También ofrece una configuración sencilla para ejecutar pruebas automatizadas tanto localmente como en entornos de integración continua (CI), permitiendo a los desarrolladores validar cambios de código de manera rápida y confiable.
- Es compatible con diversas plataformas de integración continua (CI) como GitHub Actions, GitLab CI, Travis CI, entre otras, facilitando la inclusión de pruebas automatizadas en el flujo de trabajo de desarrollo.

¿Por qué Laravel y no otros frameworks?

Laravel es la mejor opción para crear aplicaciones web modernas y completas.

- **Un marco progresivo:** Laravel crece contigo. La documentación, guías y tutoriales en vídeo te ayuda a aprender sobre Laravel. Además ofrece herramientas robustas para Inserción de dependencias, Pruebas unitarias, Colas, Eventos en tiempo real y más.
- **Un marco escalable:** Laravel es increíblemente escalable. Gracias a la naturaleza amigable de escalado de PHP y al soporte integrado de Laravel para sistemas de caché rápidos y distribuidos como Redis.
- **Un marco comunitario:** Laravel combina los mejores paquetes del ecosistema PHP para ofrecer el framework más robusto y amigable para los desarrolladores disponible.

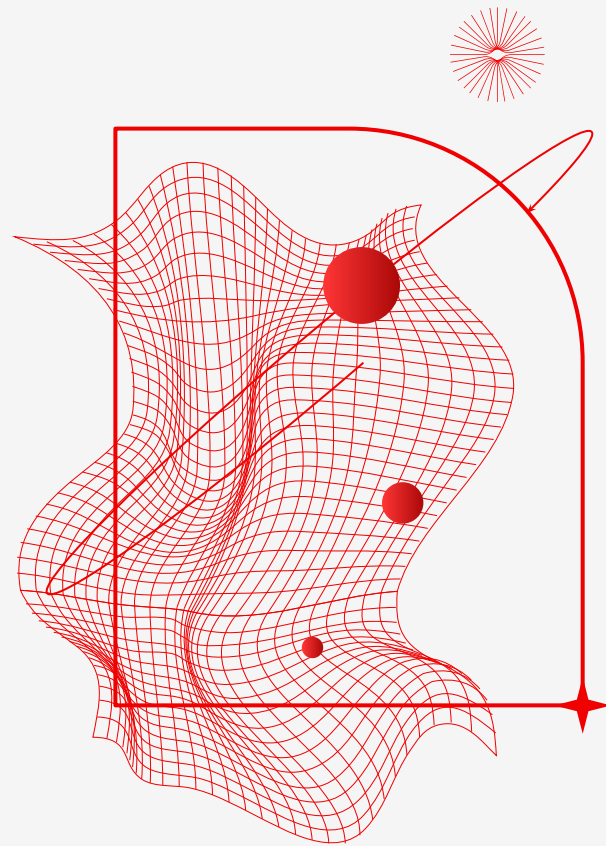
✦ Características

- **Modularidad:** Cada módulo está integrado con el administrador de dependencias de Composer, lo que facilita las actualizaciones.
- **Generador de esquemas:** Schema Builder mantiene las definiciones de la base de datos y el esquema en código PHP.
- **Correo electrónico:** Laravel incluye una clase de correo que ayuda a enviar correo con contenido enriquecido y archivos adjuntos desde la aplicación web.
- **Colas:** Laravel incluye servicios de cola como el envío de correos electrónicos a un gran número de usuarios o un trabajo Cron específico.
- **Autenticación:** Laravel facilita el diseño de la autenticación, ya que incluye funciones como registrarse, olvidar mi contraseña, etc.
- **Redis:** Laravel utiliza Redis para conectarse a una sesión existente y a una caché de uso general.



03

Arquitectura



Arquitectura y Componentes Clave

Construido sobre una arquitectura MVC, lo que facilita una separación clara entre la lógica de la aplicación y la interfaz de usuario.



Sistema de Rutas: Laravel ofrece una API muy expresiva para definir rutas. Las rutas se pueden cerrar fácilmente con middleware para agregar capas de seguridad y se pueden agrupar para organizar mejor el enrutamiento de la aplicación.

- **Middleware:** Añade una capa de lógica que se ejecuta antes o después de una solicitud, como autenticación, validación, etc.

Ejemplo:

Verificación de edad antes de acceder a una ruta

- **Agrupación de Rutas:** Permite organizar rutas con configuraciones comunes (middleware, prefijos, nombres de espacio) de manera limpia y mantenible.

Ejemplo:

Grupo de rutas que requieren autenticación y tienen un prefijo común

Arquitectura y Componentes Clave



Eloquent ORM: Proporciona una capa de abstracción para trabajar con bases de datos que permite realizar operaciones de base de datos en forma de objetos PHP.

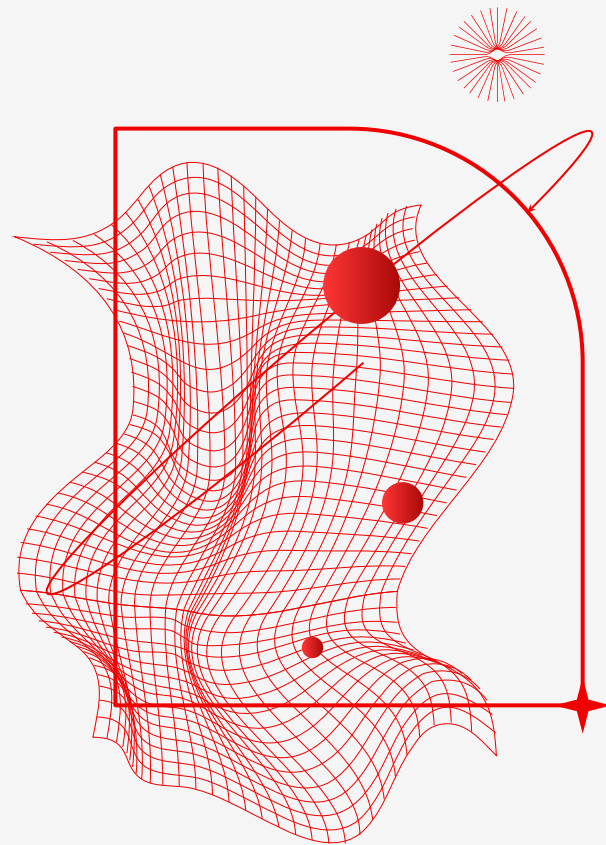


Blade Template Engine: Blade es el motor de plantillas simple pero poderoso de Laravel. Brinda a los desarrolladores la oportunidad de crear diseños complejos de forma sencilla y con un rendimiento impresionante, gracias a su naturaleza compilada.



04

Patrones de Diseño



Típicos Patrones de Diseño en Laravel



Patrón Factory

Se utiliza para construir objetos sin mostrar al cliente el mecanismo de creación. Puede utilizar el patrón de fábrica para generar nuevos objetos con datos aleatorios. Por ejemplo, puede desarrollar una clase llamada "UserFactory" que genere objetos de usuario completamente nuevos con nombres y direcciones de correo electrónico arbitrarios.



Patrón Singleton

Para asegurarse de que solo se produzca y utilice una instancia de una clase a lo largo del programa, se utiliza el enfoque singleton. Al administrar recursos como conexiones de bases de datos que no deben duplicarse, se puede utilizar el paradigma singleton de Laravel.

Típicos Patrones de Diseño en Laravel



Patrón Repository

Este diseño se utiliza para desacoplar la lógica de la aplicación de la capa de base de datos. Con Laravel, puede aprovechar el patrón de repositorio para cambiar las implementaciones de la base de datos más fácilmente sin cambiar la lógica de la aplicación.



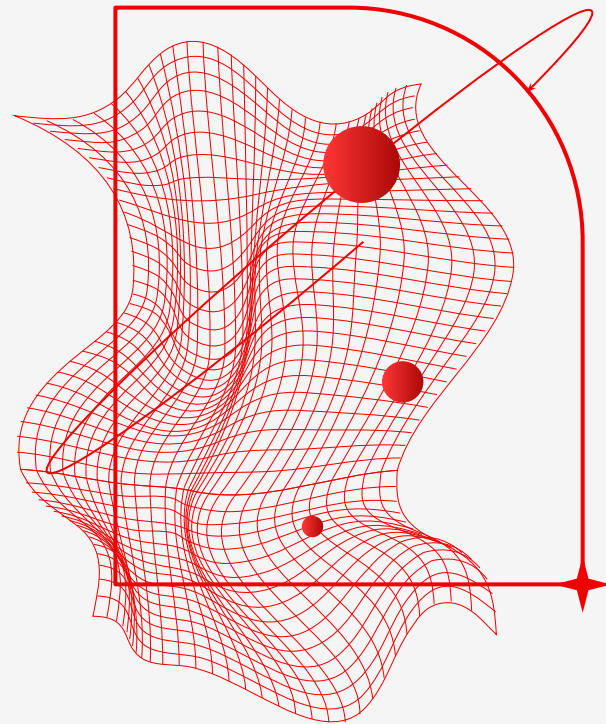
Patrón Observer

Se utiliza para desencadenar eventos cuando se llevan a cabo actividades específicas. Cuando se producen determinados eventos del modelo, puede realizar acciones adicionales utilizando el enfoque de observador de Laravel.



05

Curva de Aprendizaje



Factores Claves que Afectan la Curva de Aprendizaje

Conocimiento de Php:

- Comprender los fundamentos de PHP, como la programación orientada a objetos, funciones, arrays, y la manipulación de bases de datos con SQL, es esencial antes de comenzar con Laravel.
- El conocimiento de herramientas avanzadas como PHP Composer para la gestión de dependencias es fundamental.

Arquitectura MVC:

- **Modelos:** Representan la capa de datos y manejan las interacciones con la base de datos mediante Eloquent ORM.
- **Vistas:** Se ocupan de la presentación de los datos y están gestionadas por el motor de plantillas Blade.
- **Controladores:** Manejan la lógica de la aplicación y responden a las solicitudes del usuario.

Factores Claves que Afectan la Curva de Aprendizaje

Documentación y Recursos Educativos:

- Laravel ofrece una documentación extensa y bien organizada que cubre todos los aspectos del marco, desde la instalación hasta las características avanzadas.
- Plataformas como Laracasts proporcionan tutoriales en video y guías paso a paso que pueden acelerar significativamente el proceso de aprendizaje

Comunidad y Ecosistema:

- La comunidad de Laravel es muy activa y puede ser un recurso valioso para resolver problemas y obtener consejos.
- Laravel tiene un ecosistema robusto de paquetes que pueden extender la funcionalidad de las aplicaciones, lo que puede simplificar muchas tareas comunes pero también agregar complejidad inicial.

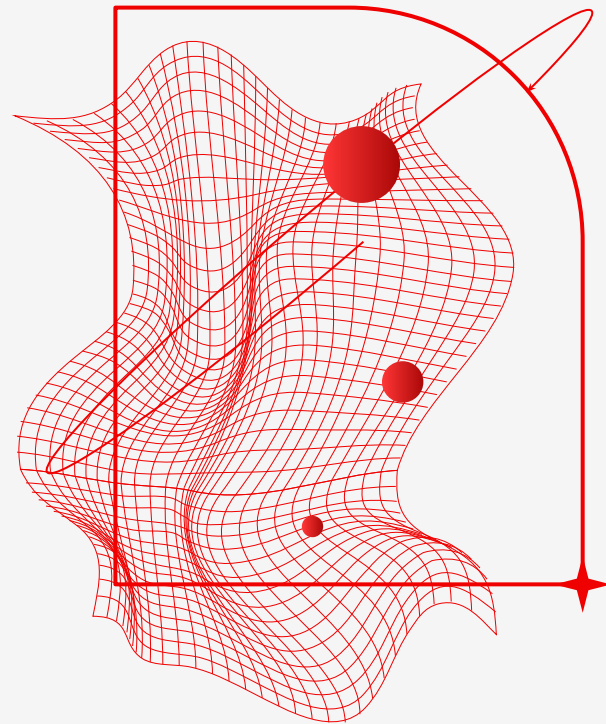
Comparación con Otros Marcos de Trabajo

Laravel es conocido por tener una curva de aprendizaje más amigable en comparación con otros marcos de PHP, debido a su sintaxis limpia y diseño opinado que guía a los desarrolladores hacia las mejores prácticas. Sin embargo, para los desarrolladores completamente nuevos en el desarrollo web o en PHP, puede haber una curva de aprendizaje inicial pronunciada.



06

Referencias



Referencias Bibliográficas

- LaRavel - the PHP framework for web artisans. (s. f.). <https://laravel.com/docs/11.x/installation>
- Cabrera, A. (2024, febrero 27). Qué es Laravel y Cómo puede Transformar el Desarrollo de Tus Aplicaciones Web. 10Code Software Design. <https://10code.es/laravel-que-es/>
- Rasool, G. (2023, julio 10). Typical design patterns for Laravel application development. Medium. <https://medium.com/@imgrasool/typical-design-patterns-for-laravel-application-development-b97f1ee5fd6b>
- divcode. (2023, 12 mayo). *como INSTALAR LARAVEL en VISUAL STUDIO CODE* 🤖 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=rltpKZosln4>
- Cloudways. (n.d.). Grasping the Laravel Development Learning Curve – A Simple Guide. Recuperado de <https://www.cloudways.com/blog/laravel-learning-curve>
- DZone. (2024, enero 10). Laravel for Beginners: An Overview of the PHP Framework. Recuperado de <https://dzone.com/articles/laravel-for-beginners-an-overview-of-the-php-framework>
- Laracasts. (n.d.). Laravel Learning Curve Discussion. Recuperado de <https://laracasts.com/discuss/channels/laravel/laravel-learning-curve-time>
-

**GRACIAS POR
SU ATENCIÓN**