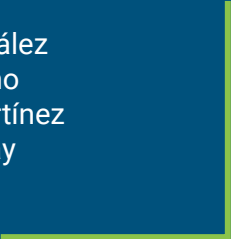




FRAMEWORKS



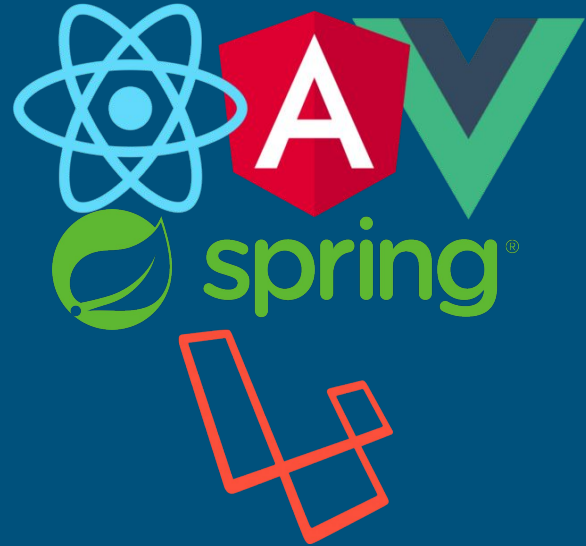
David González
Alberto Olmo
Alberto Martínez
Erika Guzñay
1ºDAW

ÍNDICE

1. ¿QUÉ ES UN FRAMEWORK?
2. ¿PARA QUÉ SIRVEN?
3. FRAMEWORKS DE FRONT-END
4. FRAMEWORKS DE BACK-END
5. DESVENTAJAS
6. VENTAJAS

1º ¿QUÉ ES UN FRAMEWORK?

- Estructura previa para facilitar el desarrollo.
- Basado en plantilla/esquema conceptual.
- Se usa en desarrollo y márketing.



2º ¿PARA QUÉ SIRVE?

- Permiten un punto de partida más avanzado.
- Evitan errores de escritura de código.
- Aportan mayor seguridad.
- Ayudan al mantenimiento y reutilización de código.
- Acortar plazos de desarrollo.

3º FRONT-END (ENTORNO CLIENTE)

- Se centra en la **visualización** del contenido y la **interacción** con el usuario final.
- Desarrolla las **UI** (interfaces de usuario) y trabaja la **UX** (experiencia de usuario).
- Permite:
 - Actualización de **cambios** en **tiempo real**.
 - Aplicación de **directivas** (operaciones complejas de forma sencilla).
 - Uso de **plantillas**.
- Algunos de los **más relevantes** son:
 - Angular
 - Vue.js
 - Bootstrap



4º BACK-END (ENTORNO SERVIDOR)



Symfony



django



- Se encarga de llevar la información al usuario.
- Se ocupa de diseñar la lógica y las soluciones para que las acciones ejecutadas sean realizadas correctamente.
- Diríamos que es la **parte no visible** (servidor, base de datos, aplicaciones).

5º DESVENTAJAS

- Exceso de líneas de código (ya que suelen incluir “código basura”).
- Limitaciones: hay partes de él que no puedes modificar.
- Código público: puede ser menos seguro.

6º VENTAJAS

- Autenticación: permite identificarte como usuario.
- Internalización: permite mostrar la aplicación en diferentes idiomas.
- Acceso a datos: incluyen herramientas e interfaces que te permiten comunicarte con la base de datos.
- Uso de controladores: sirven para la gestión de los eventos y peticiones realizadas a la aplicación.