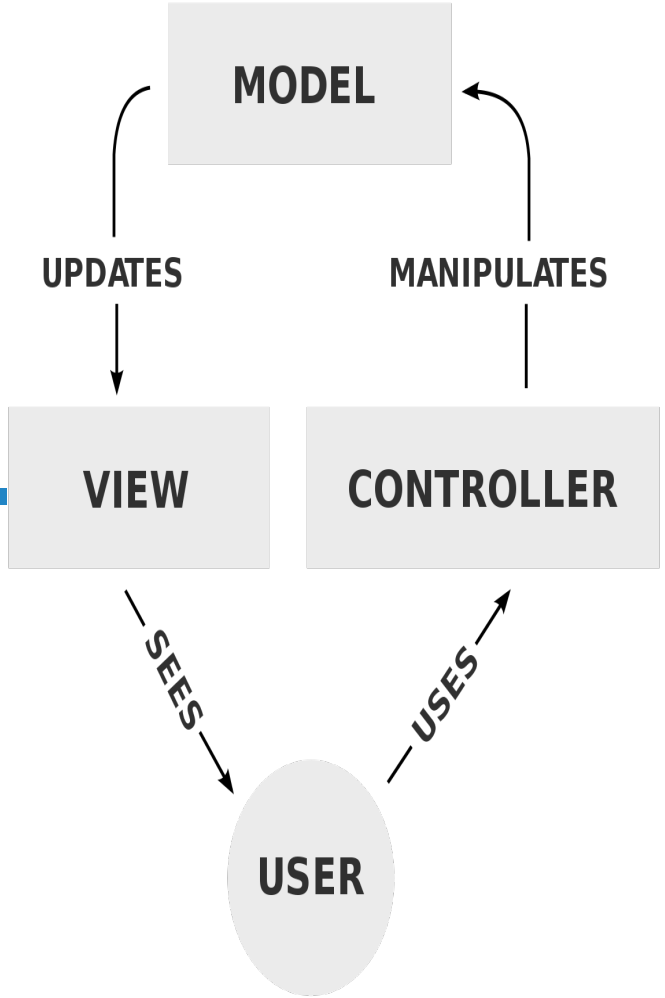
UNIVERSIDAD D CÓRDOBA

PROGRAMACIÓN WEB – BLOQUE II

Fundamentos del desarrollo Web



Dr. José Raúl Romero Salguero
jrromero@uco.es

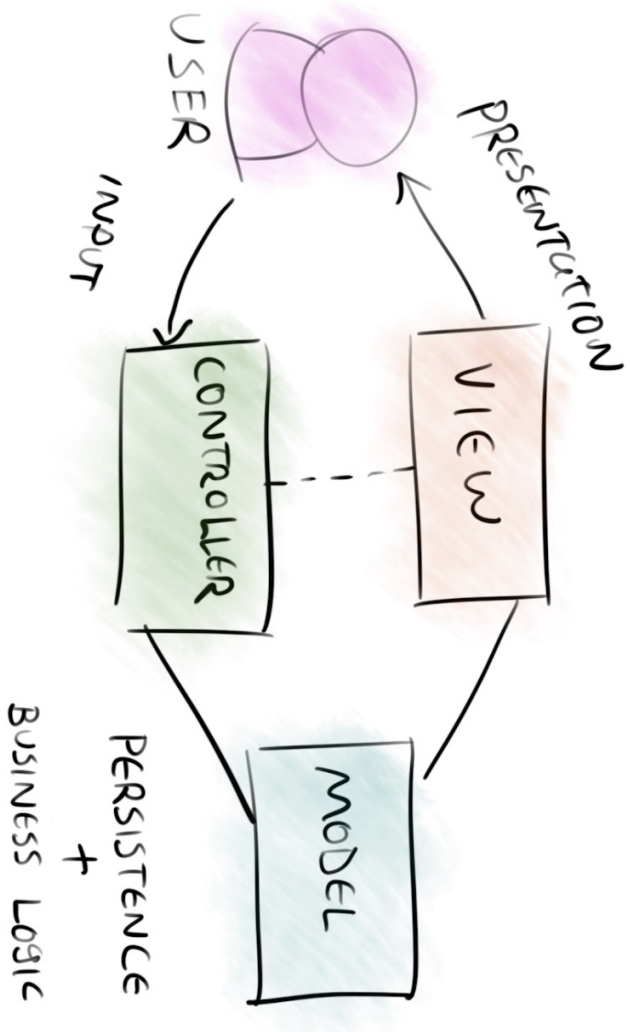
Contenidos del Bloque

1. Marcos tecnológicos
2. Lenguajes para la web
3. Principios de diseño y arquitectura

UNIVERSIDAD D CÓRDOBA

PROGRAMACIÓN WEB – TEMA II-1

Marcos tecnológicos



Dr. José Raúl Romero Salguero
jrromero@uco.es

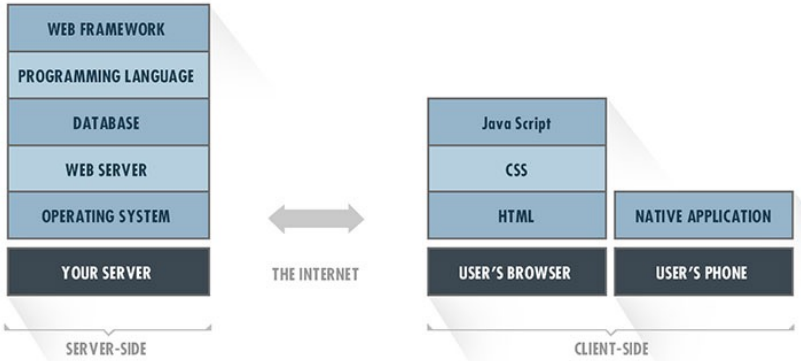
Contenidos

1. Pila de tecnología
2. Web frameworks

1.
Pila de tecnología

Pila de tecnología

- En desarrollo web, el término "pila de tecnología" (***tech stack***) se refiere al conjunto de herramientas, lenguajes y software que se emplea durante las fases de desarrollo y despliegue de un producto web o aplicación móvil
- En general, la pila se compone de la combinación de tecnologías utilizadas en el lado del servidor (*back-end*) y en el lado del cliente (*front-end*)
- *Ejemplo de pila*: LAMP (Linux + Apache + MySQL + PHP)



Fuente: Silicon Valley Software Group

Pila de tecnología__
Back-end Vs. *Front-end*

- El *back-end* contiene la lógica de negocio de la aplicación
- El *back-end* nunca es accedido por el usuario directamente, sino a través del *front-end*
- El *back-end* se desarrolla con lenguajes de programación, según las necesidades del proyecto (Java, Ruby, PHP, etc.)

- El *front-end* es la parte visual de la aplicación con la interactúa el usuario final
- El *front-end* se desarrolla predominantemente con lenguajes de marcado (HTML, CSS) y el lenguaje de programación Javascript

2.

Web frameworks

¿Qué es un "*Framework*"?

- Un *framework* es un software abstracto, que ofrece funciones genéricas para ser utilizadas o adaptadas por el código del desarrollador y facilitar así el desarrollo de aplicaciones específicas
  - Para considerarse *framework*, debe ser independiente del dominio de aplicación, universal y ofrecer un entorno reutilizable
  - Puede ofrecer soporte adicional, como herramientas, APIs, librerías de código, etc.
- Características distintivas:
  - Inversión de control: el flujo del sistema lo controla el *framework* (vs. APIs)
  - Extensibilidad: El desarrollador podrá extender su funcionalidad
  - Estabilidad: El código del *framework* es transparente al desarrollo, que no preverá cambios en el código que afecten a su aplicación
  - Compatibilidad hacia atrás: Debe garantizar la compatibilidad de la aplicación conforme evoluciona (p.ej. por motivos de seguridad)

¿Qué es un "*Web Framework*" (WF)?

- Un *web framework* (WF) es un *framework* software diseñado para el desarrollo de aplicaciones web a través de invocaciones a servicios web, APIs u otro tipo de recursos
- El objetivo principal del WF es el de **automatizar** partes del desarrollo web (p.ej. patrón MVC) y otras actividades comunes (p.ej. acceso a base de datos) que –de otra forma– deberían ser implementadas una y otra vez
  - Facilitan la reutilización de código
- Generalmente se utilizan para el desarrollo de sitios web dinámicos
- Los WF se diferencian según el lado en que se despliegan: WF de *back-end* y WF de *front-end*

WF: Características

- Características habituales de los WF:
  - Sistema basado en plantillas (*templates*): Mecanismos para la generación automática de documentos web a partir de un lenguaje de marcas – favoreciendo reutilización de código
  - Gestión de caché: Almacenamiento temporal de documentos, así como su actualización y gestión, con el fin de reducir el ancho de banda y los accesos al servidor de aplicaciones (mejorando la latencia)
  - Acceso a base de datos y gestión de objetos de datos: Acceso a un amplio número de bases de datos sin implicar modificación en el código fuente de la aplicación, así como la creación de objetos de mapeo y correspondencia con el esquema de la base de datos (mediante objetos de clases o formatos tipos JSON o XML)

WF: Características

- Funciones de seguridad: Funciones de autenticación y autorización, así como listas de control de acceso a las funciones de la aplicación
- Enrutado: Mecanismos para la interpretación, mapeo y redirección de URLs (*friendly URL*)
- Programación asíncrona: Encapsulan la dificultad de las técnicas de desarrollo de comunicación asíncrona con el servidor (*AJAX – Asynchronous JavaScript and XML*)
- Generación de código: Tanto a nivel de patrones arquitecturales (MVC) –incluyendo acceso a base de datos– como a nivel de plantillas
- Servicios web: Encapsulación de creación y suministro de WS
- Integración: Mecanismos de integración con otros *frameworks* o herramientas (p.ej. con *frameworks* de pruebas tipo JUnit)

WF: Criterios de elección

- Un objetivo importante en el desarrollo web es reducir el *time-to-market*
- Debido a la gran variedad de WF, Mozilla recomienda seguir una serie de criterios de elección:
  - Curva de aprendizaje: Tiempo requerido para dominar las funcionalidades del WF
  - Productividad: Cómo de rápido se desarrolla con el *framework*, así como el esfuerzo de mantenimiento del código.
  - Uso de buenas prácticas: Referido a si el WF potencia el uso de "buenas prácticas", como patrones (ej. MVC) o capacidad de prueba (ej. implementación de tests)

WF: Criterios de elección

- Desempeño del WF: Tiempo de ejecución y carga del sitio. En general, este factor puede compensarse de otras formas en el despliegue (máquinas, balanceadores, proxy inverso u otro hardware o software base)
- Escalabilidad: Tanto escalabilidad vertical (cambiar a un hardware más potente) como escalabilidad horizontal (ampliar el número de servidores web y bases de datos) y/o geográfica (redistribuir los usuarios según su localización más cercana a un servidor)
- Soporte de caché: Permite soportar un mayor número de peticiones optimizando el documento (o parte del mismo) que se guarda la primera vez que se solicita para no ser generado de nuevo
- Seguridad web: Resistencia frente a ataques y frecuencia de actualizaciones del WF al respecto

Ejemplos de WF de *back-end*

Web framework	Lenguaje	Se caracteriza por…
Ruby on Rails	Ruby	Orientado a datos con MVC. Extremadamente productivo (x10 respecto a WF de Java). Ejemplos: Groupon, Github, AirBnb
Django	Python	Orientado a aplicaciones de medios y publicación de usuarios (de uso generalista). Altamente escalable y rápido (¡Python no lo es tanto!). Ejemplos: Disqus, Pinterest, Instagram
ASP.net	C#	Orientado a aplicaciones industriales robustas y dispositivos móviles. Ligero y con alto rendimiento dentro .NET. Ejemplos: StackOverflow, GettyImages
CodeIgniter	PHP	Débilmente acoplado basado en MVC. Alto rendimiento, baja/cero-configuración y sin grandes librerías monolíticas. Ejemplo: The Guardian.
Spring	Java Enterprise	Orientado a aplicaciones empresariales que requieren robustez, seguridad, alto rendimiento, portabilidad e integración con ecosistema JVM. Ejemplo: MIT, Ticketmaster, Fitbit.
Play	Scala, Java	Orientado a optimizar la productividad del desarrollador frente a configuraciones complejas, recarga de código y control de errores. Ejemplos: LinkedIn, Coursera

- Estos WF se basan en tecnologías Javascript, el único lenguaje de programación considerable para *front-end*
- En general, estos WF permiten desarrollar documentos ricos e interactivos de forma sencilla, escalable y mantenible
- Ejemplos actuales más significativos:
  - ANGULAR. WF de Google para el desarrollo de aplicaciones de gran tamaño y alto rendimiento, con foco en el mantenimiento. Adopta un estilo declarativo de programación (frente al imperativo) Ejemplos: YouTube (PS3), Netflix, Microsoft Office Home.
  - REACT. Librería -originaria de Facebook- basada en componentes (renderizado independiente) para la composición de la interfaz de usuario. También utilizado para aplicaciones móviles. Su objetivo es la simplicidad y facilidad de aprendizaje, testabilidad y control de los datos. Adopta un estilo reactivo de programación (estilo declarativo guiado por datos). Ejemplos: Facebook, Netflix, Whatsapp Web.

- BACKBONE.JS. Su objetivo es ofrecer facilidad para el desarrollo de SPA implementando MVC. Permite mantener sincronizados múltiples clientes con el servidor. Sigue un modo imperativo de desarrollo. Ejemplos: LinkedIn, Digg, Trello. [Tiende a librería más que WF]
- VUE.JS. Desarrollado por antiguo trabajador de Google y por comunidad. WF progresivo (desde páginas sencillas a SPAs), su objetivo es la sencillez y disminuir la curva de aprendizaje. Trabaja con componentes, implementando MVVM (model-view-viewmodel). Ejemplos: Xiaomi, Alibaba, Gitlab.

- También hay otras librerías (con herramientas asociadas) focalizadas en la creación de espacios responsivos, que facilita la incorporación de elementos estéticos en la web (p.ej. BOOTSTRAP, JQUERY, etc.) si bien no están estrictamente vinculadas a un WF.

Micro-frameworks

- Término referido a un WF minimalista, esto es, aquel que ofrece un subconjunto mínimo de funcionalidades necesario para el desarrollo de un proyecto web
- En general, un micro-WF ofrece las siguientes características mínimas:
  - Seguridad. Gestión de cuentas, autenticación y autorización
  - Validación de entradas
  - Motor de plantillas (*templates*)
  - Mapeo de objetos de acceso a datos
  - Control y gestión de solicitudes y respuestas HTTP
  - Integración con APIs (ej. aplicaciones basadas en servicios)

Programación Web

Fundamentos del desarrollo web__ **Curso 2021/22**