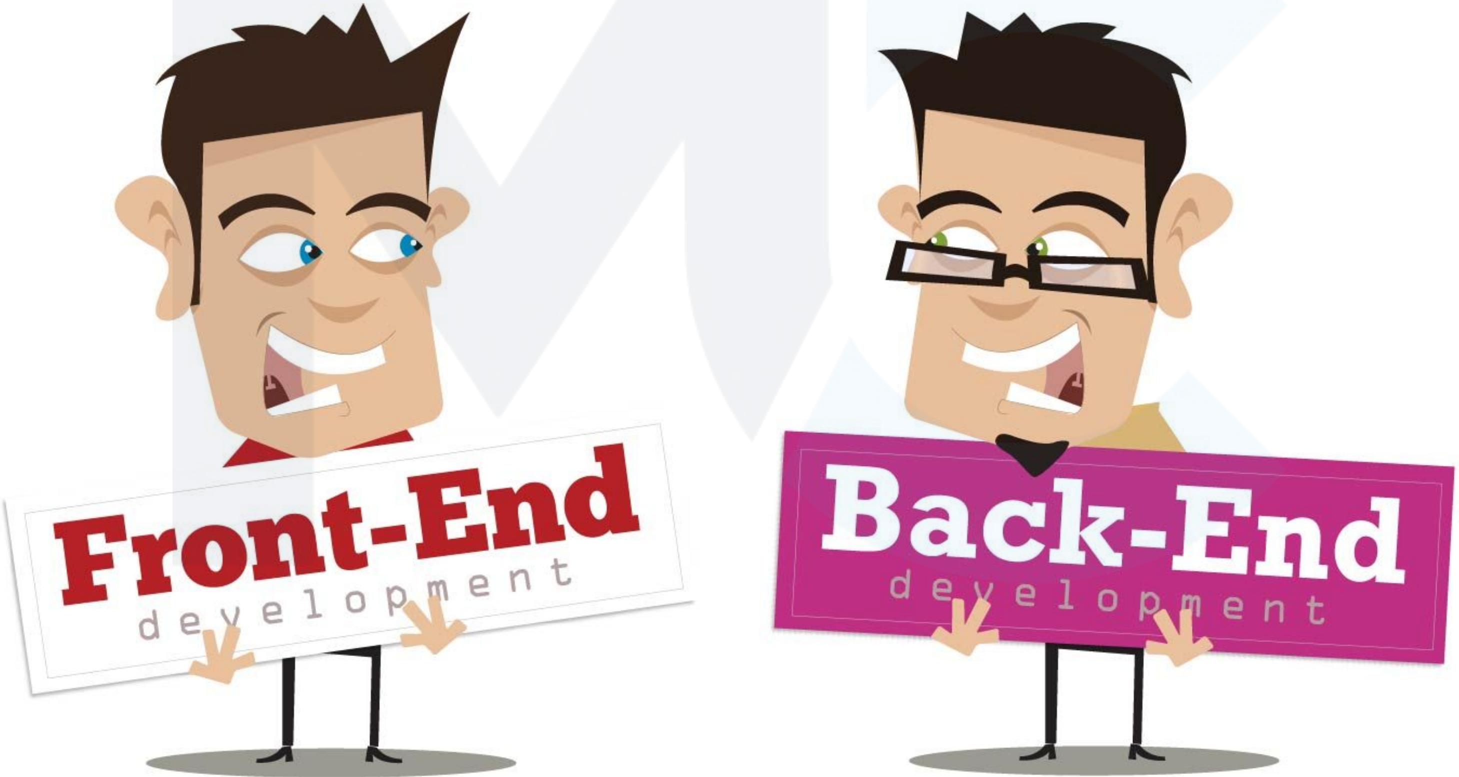
JAVA FULL STACK DEVELOPER

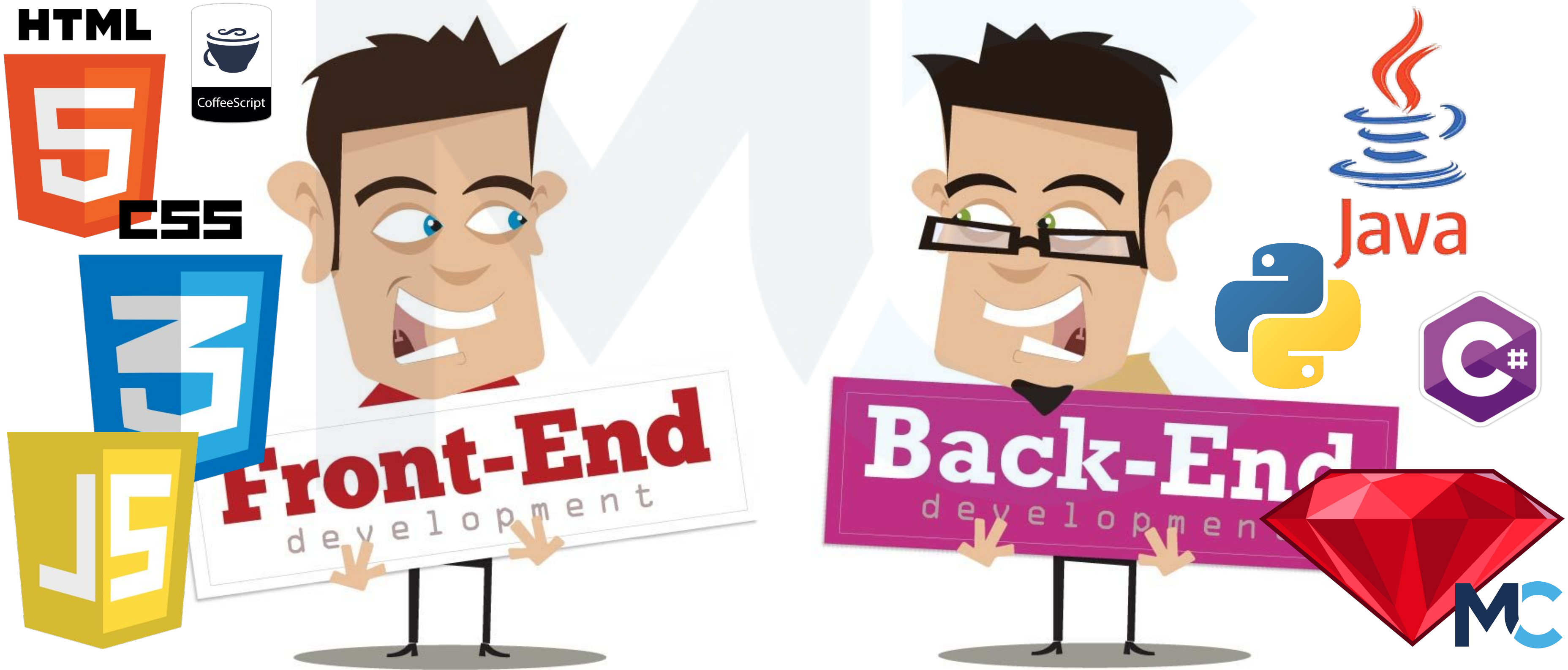
MitoCode Network

Edición 2022

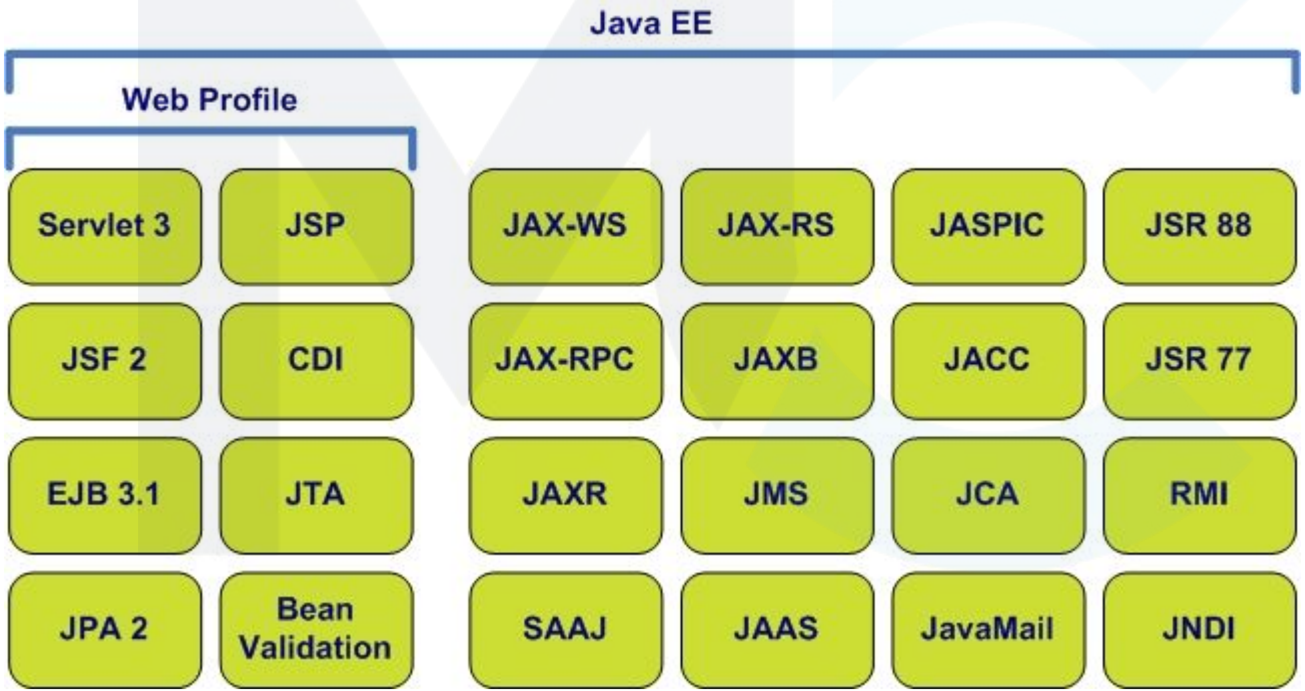
¿Back-end vs Front-end?



¿Back-end vs Front-end?



Hablemos de Java (Java EE | Jakarta)

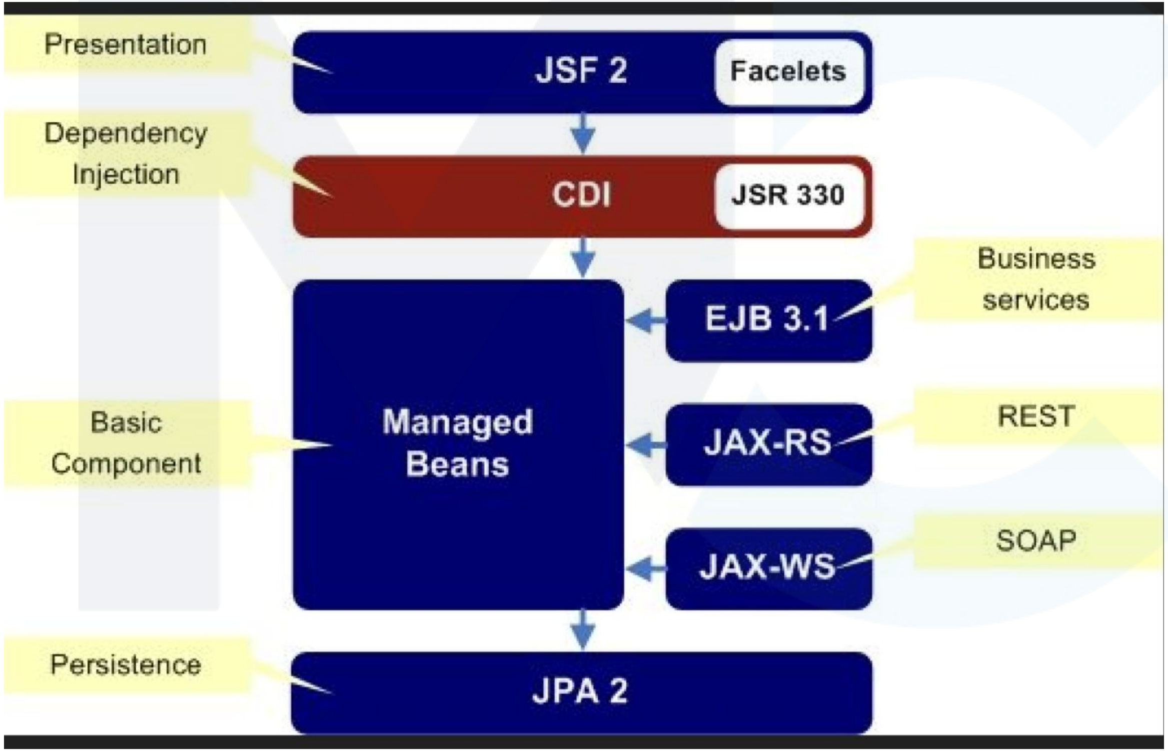


Java EE

Web Profile

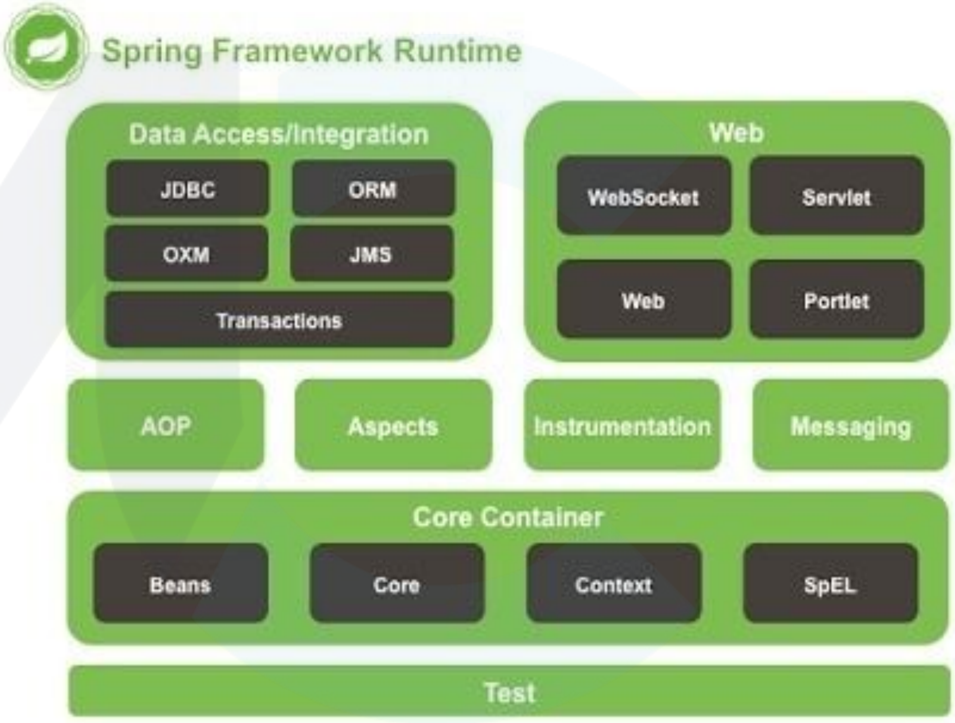
Servlet 3	JSP	JAX-WS	JAX-RS	JASPIC	JSR 88
JSF 2	CDI	JAX-RPC	JAXB	JACC	JSR 77
EJB 3.1	JTA	JAXR	JMS	JCA	RMI
JPA 2	Bean Validation	SAAJ	JAAS	JavaMail	JNDI

Hablemos de Java (Java EE | Jakarta)

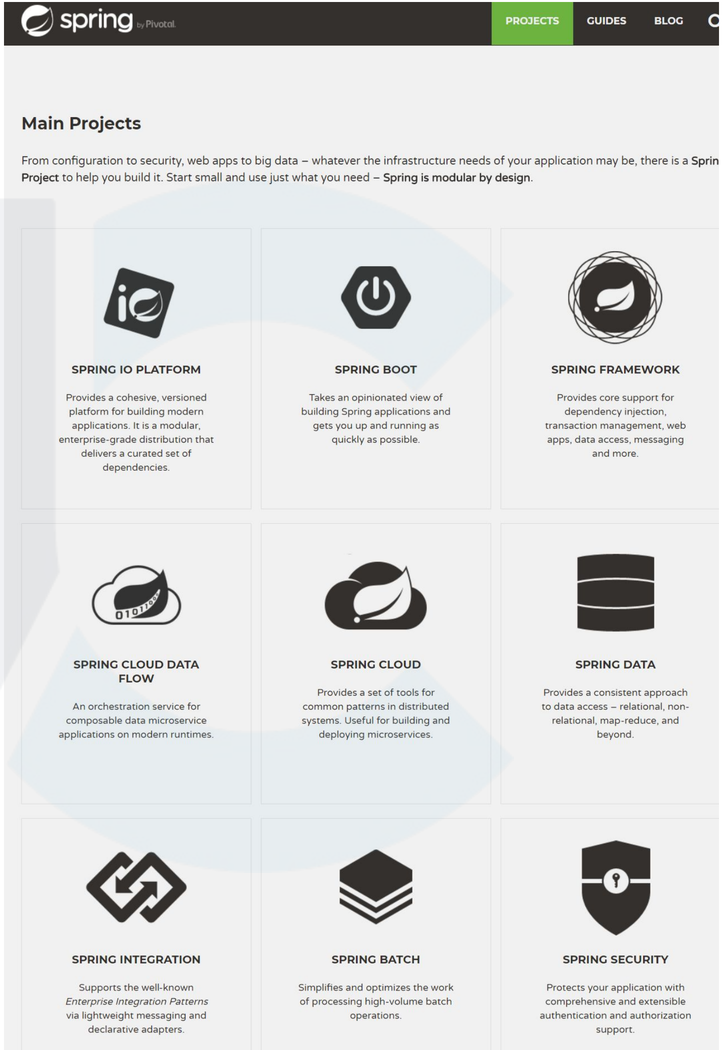


Hablemos de Spring

Spring Framework



Spring Projects



Main Projects

From configuration to security, web apps to big data – whatever the infrastructure needs of your application may be, there is a **Sprin** **Project** to help you build it. Start small and use just what you need – **Spring is modular by design**.

SPRING IO PLATFORM

Provides a cohesive, versioned platform for building modern applications. It is a modular, enterprise-grade distribution that delivers a curated set of dependencies.

SPRING BOOT

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

SPRING FRAMEWORK

Provides core support for dependency injection, transaction management, web apps, data access, messaging and more.

SPRING CLOUD DATA FLOW

An orchestration service for composable data microservice applications on modern runtimes.

SPRING CLOUD

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.

SPRING DATA

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.

SPRING INTEGRATION

Supports the well-known *Enterprise Integration Patterns* via lightweight messaging and declarative adapters.

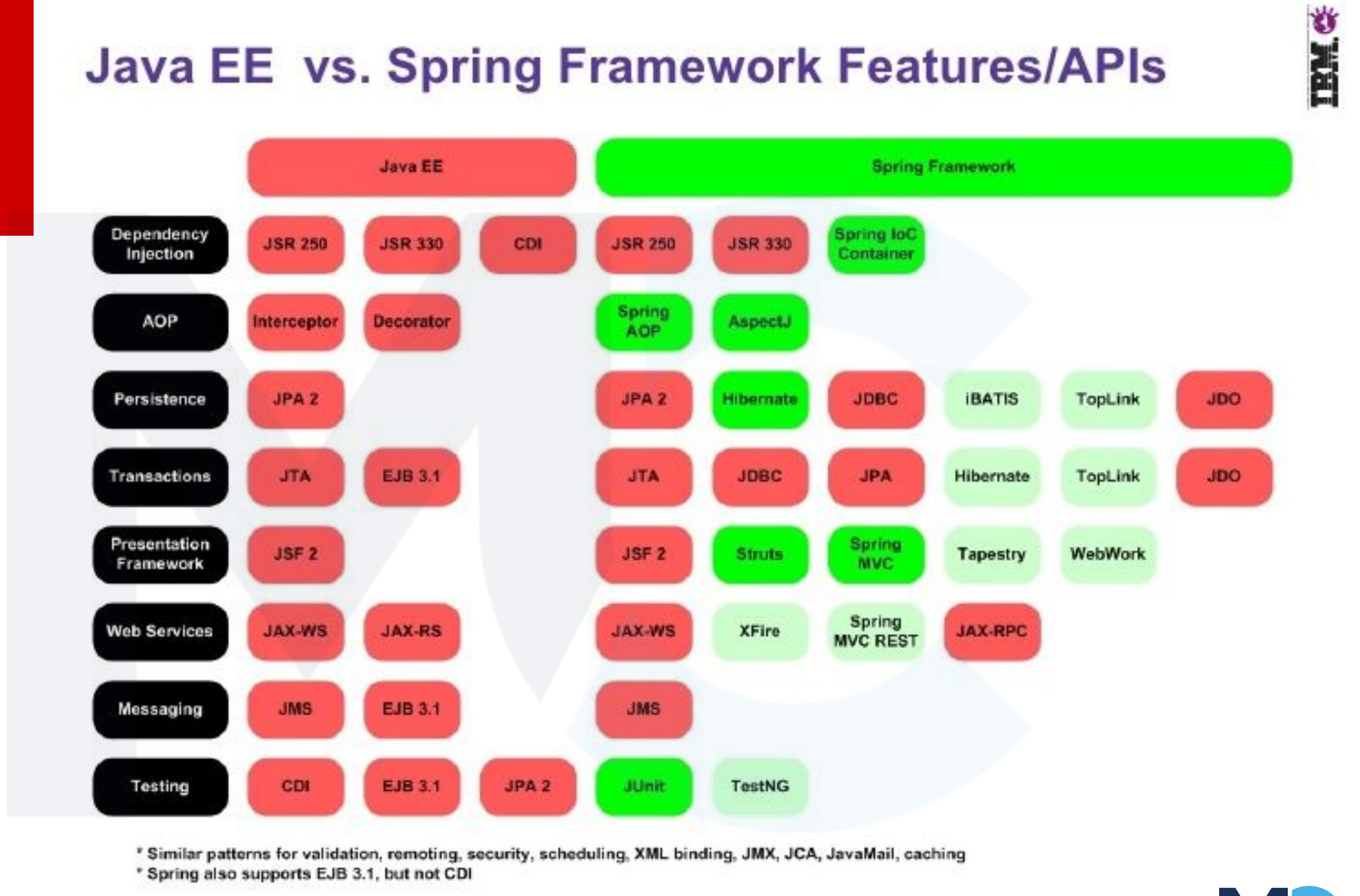
SPRING BATCH

Simplifies and optimizes the work of processing high-volume batch operations.

SPRING SECURITY

Protects your application with comprehensive and extensible authentication and authorization support.

Comparativa



Java EE vs. Spring Framework Features/APIs

	Java EE			Spring Framework					
Dependency Injection	JSR 250	JSR 330	CDI	JSR 250	JSR 330	Spring IoC Container			
AOP	Interceptor	Decorator		Spring AOP	AspectJ				
Persistence	JPA 2			JPA 2	Hibernate	JDBC	iBATIS	TopLink	JDO
Transactions	JTA	EJB 3.1		JTA	JDBC	JPA	Hibernate	TopLink	JDO
Presentation Framework	JSF 2			JSF 2	Struts	Spring MVC	Tapestry	WebWork	
Web Services	JAX-WS	JAX-RS		JAX-WS	XFire	Spring MVC REST	JAX-RPC		
Messaging	JMS	EJB 3.1		JMS					
Testing	CDI	EJB 3.1	JPA 2	JUnit	TestNG				

* Similar patterns for validation, remoting, security, scheduling, XML binding, JMX, JCA, JavaMail, caching
* Spring also supports EJB 3.1, but not CDI

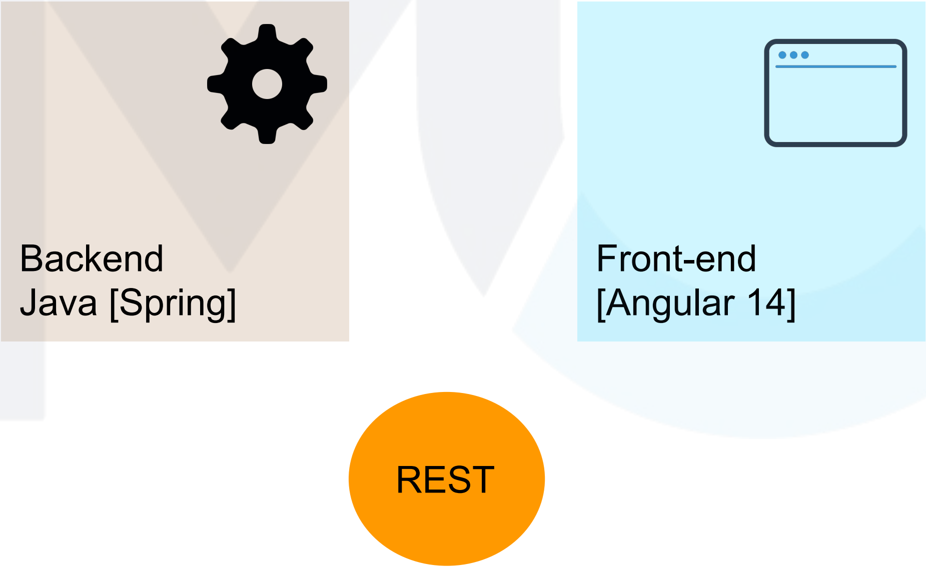
Fuente: https://www.slideshare.net/kelapure/2012-0406v2tdp1163java-eevsspringshootoutfinal.

¿Dónde nos centraremos en el curso?

Backend
Java [Spring]

Front-end
[Angular 14]

¿Dónde nos centraremos en el curso?

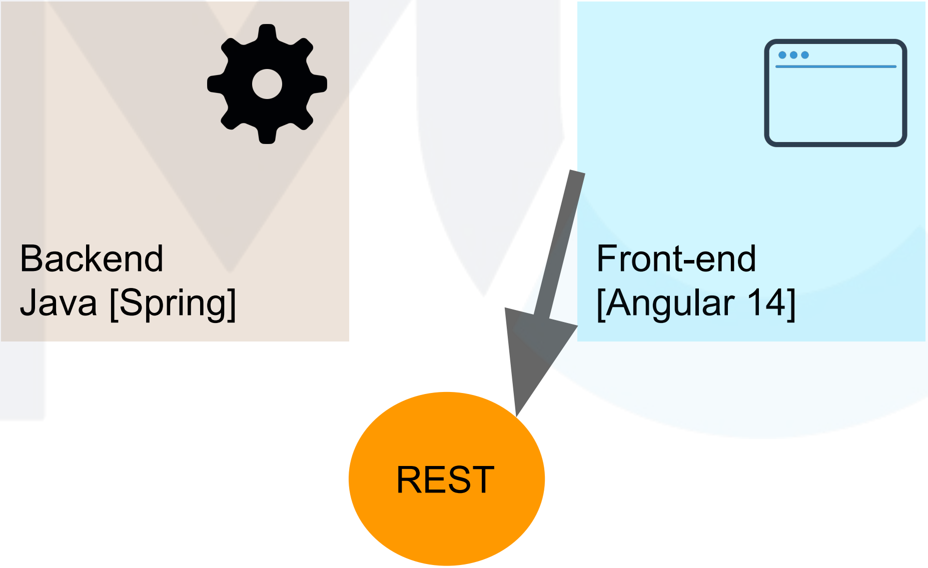


Backend
Java [Spring]

Front-end
[Angular 14]

REST

¿Dónde nos centraremos en el curso?

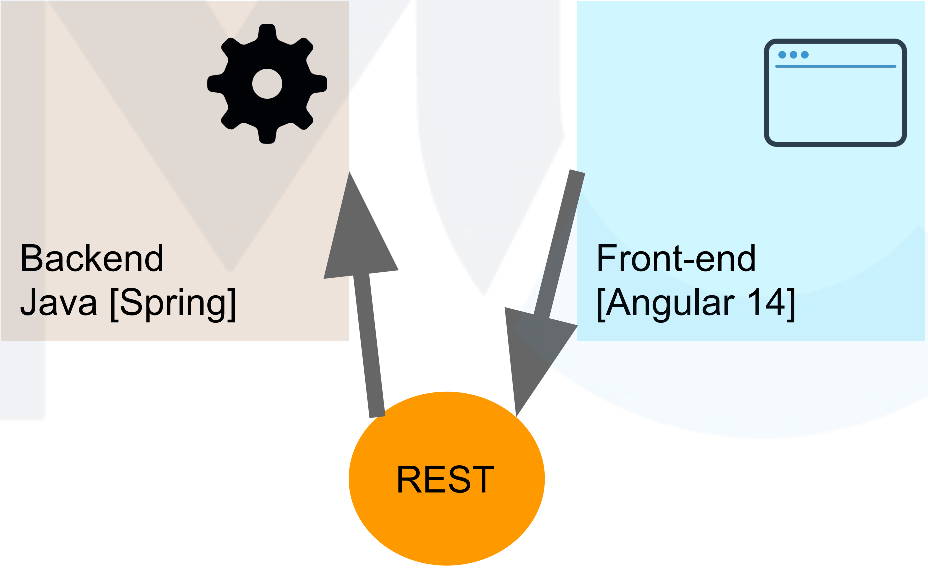


Backend
Java [Spring]

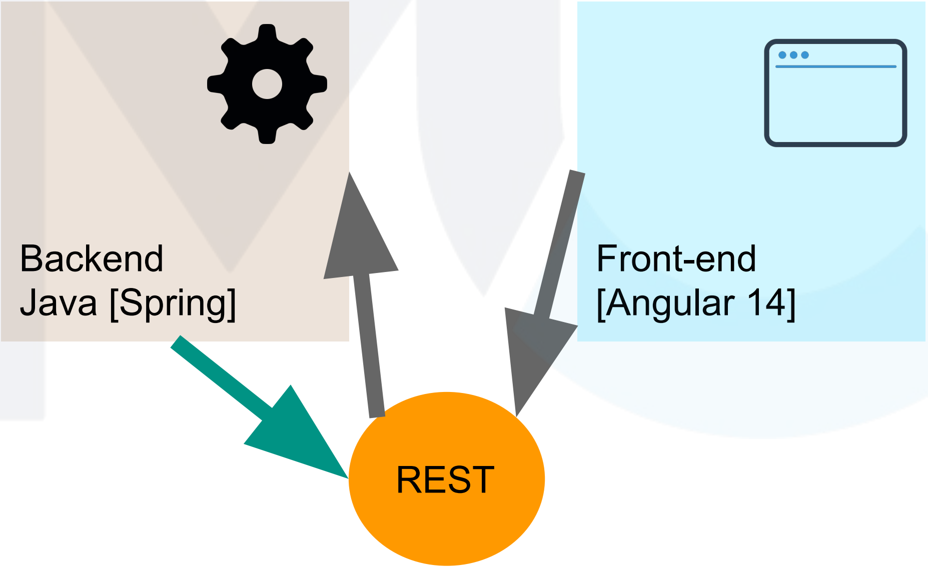
Front-end
[Angular 14]

REST

¿Dónde nos centraremos en el curso?



¿Dónde nos centraremos en el curso?

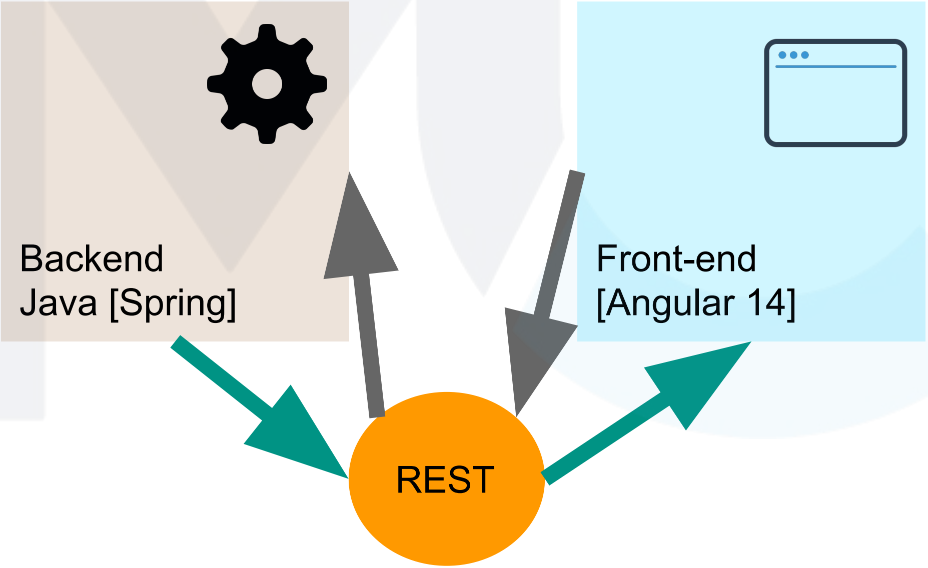


Backend
Java [Spring]

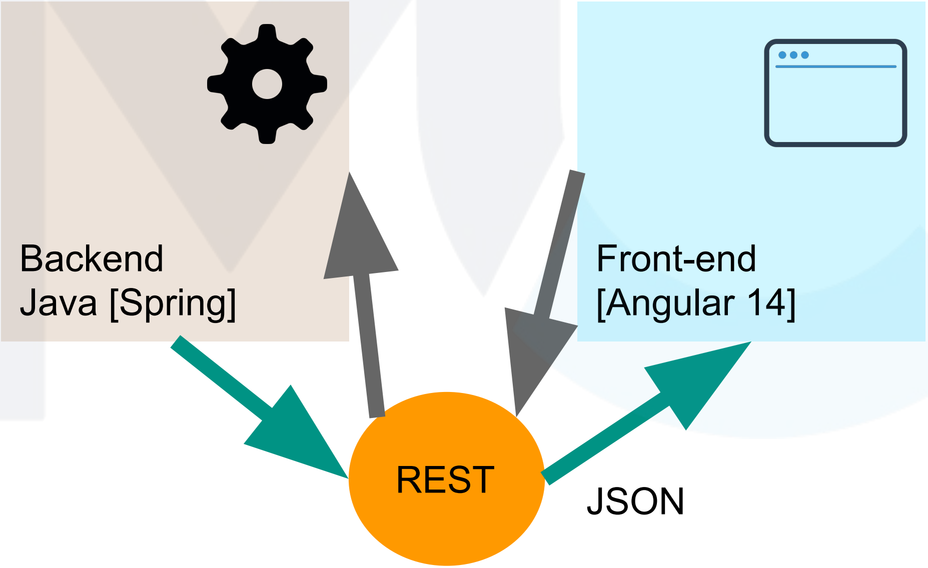
Front-end
[Angular 14]

REST

¿Dónde nos centraremos en el curso?



¿Dónde nos centraremos en el curso?



Backend
Java [Spring]

Front-end
[Angular 14]

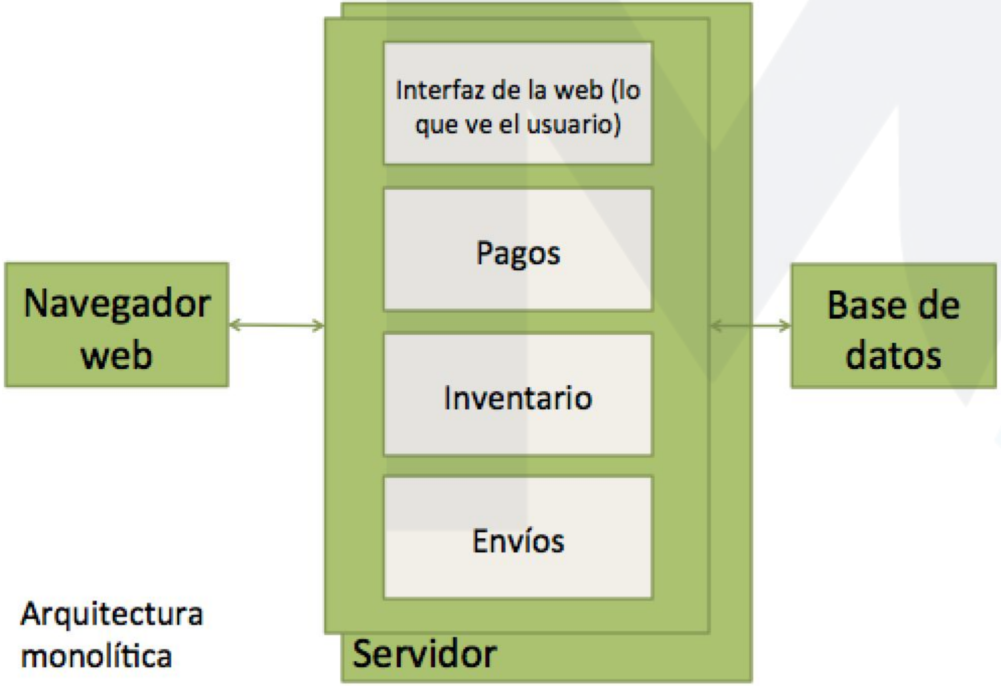
REST

JSON

Objetivos del curso

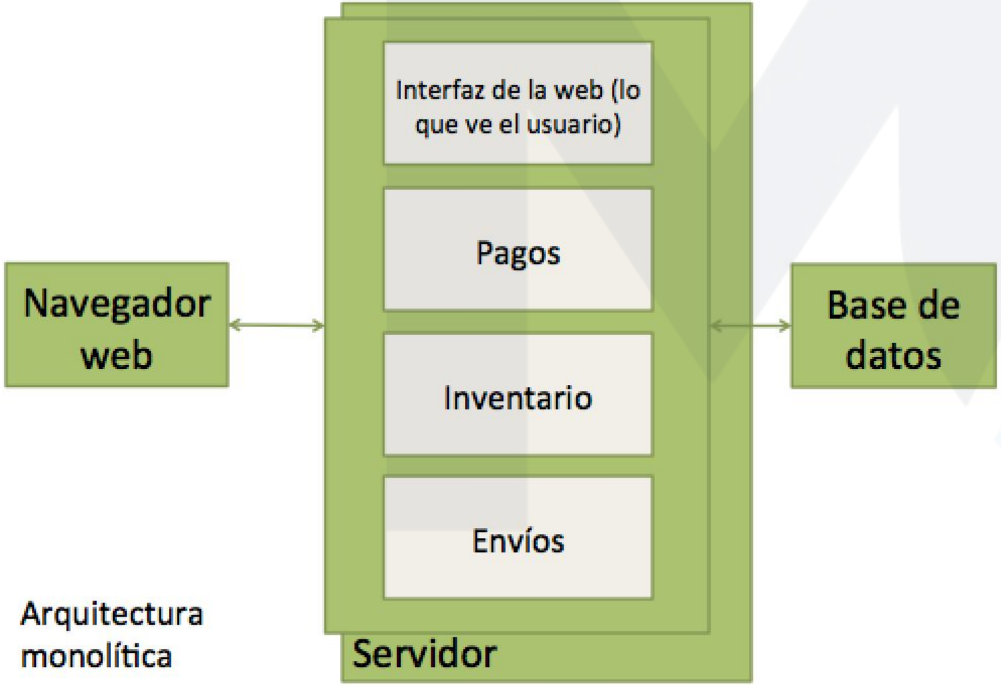
- Crear un backend con Spring [Boot]
  - Servicios REST
  - Protección por JWT
  - Spring Data JPA Repository
  - JSON
  - Enfoque Monolítico
  - Enfoque Microservicios
- Crear un frontend con Angular 14
  - Angular Material
  - Seguridad y comunicación con servicios
- Puesta en producción.

Arquitectura 1 [Monolito]



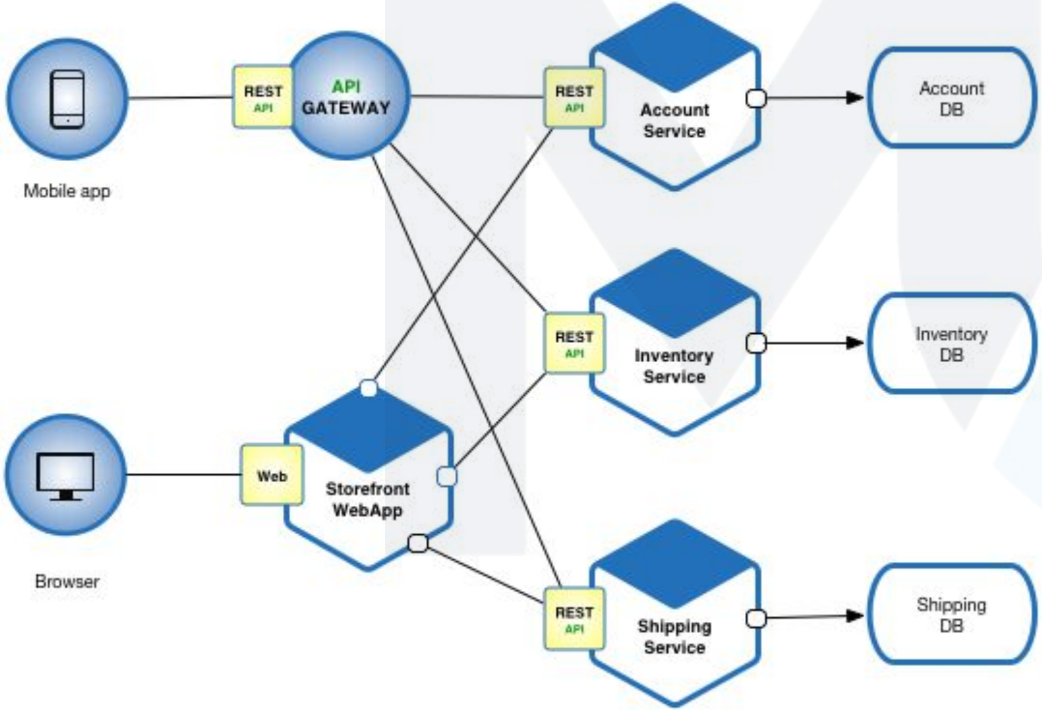
Arquitectura monolítica

Arquitectura 1 [Monolito]

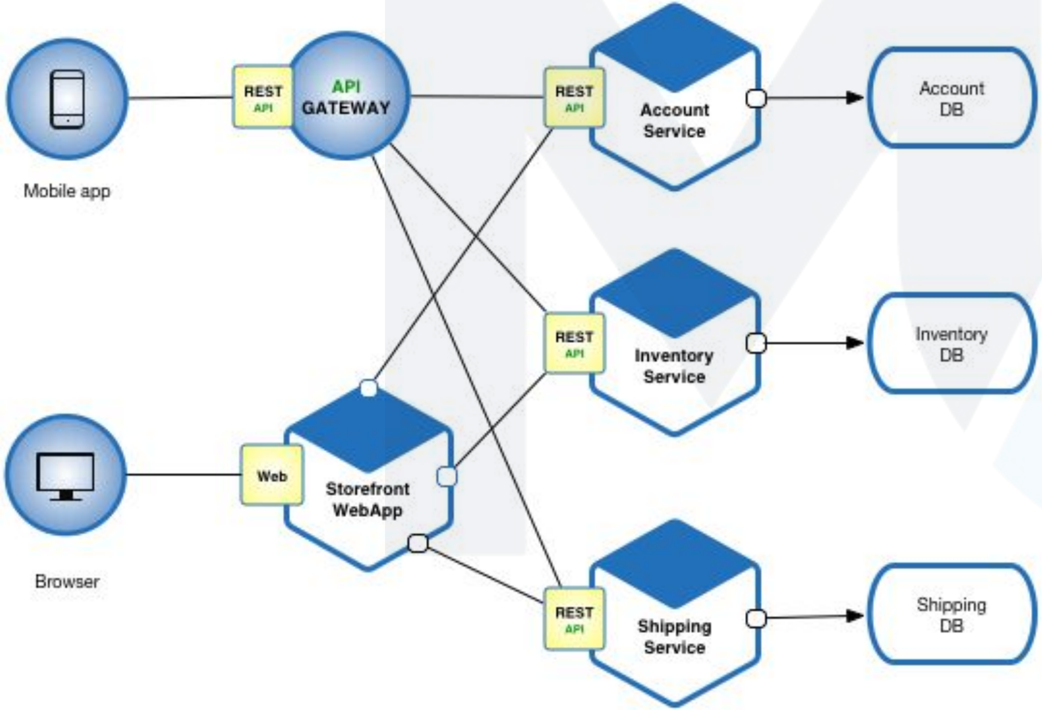


- Simple desarrollo
- Simple de probar
- Simple de desplegar
- Simple de escalar

Arquitectura 2 [Microservicios]

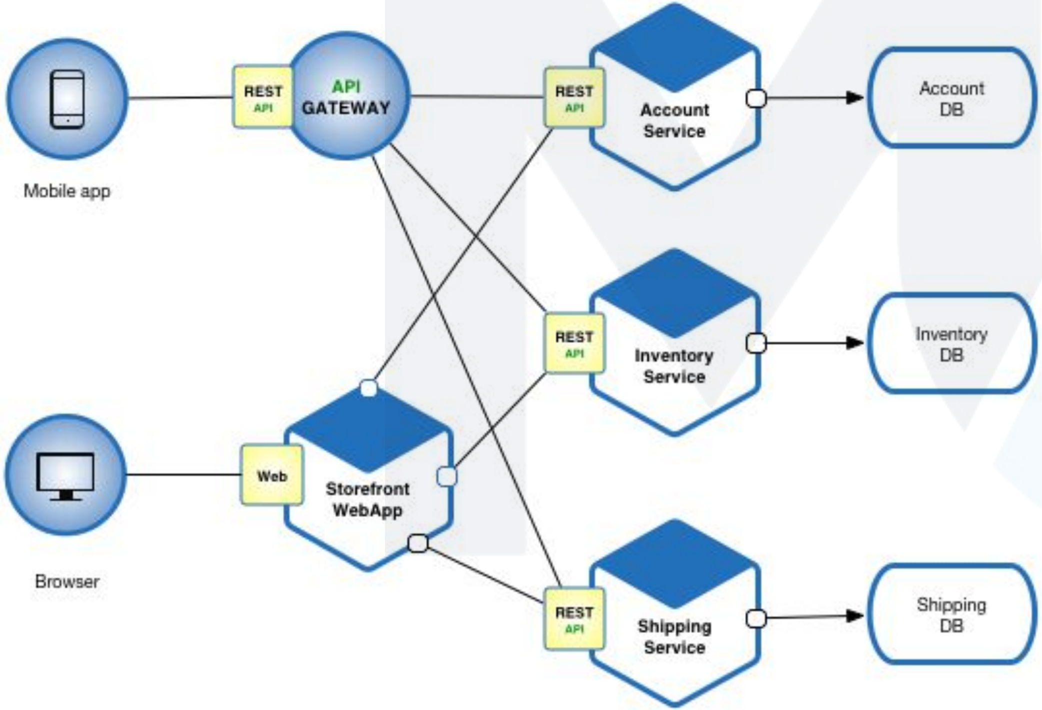


Arquitectura 2 [Microservicios] – Ventajas



- Simple mantenimiento
- Equipo multi-lenguaje (diversidad tecnológica)
- Desarrollos en paralelo

Arquitectura 2 [Microservicios] – Desventajas



- Complejidad de configuración
- Complejidad de seguridad
- Complejidad de gestionar errores

¡Empecemos!