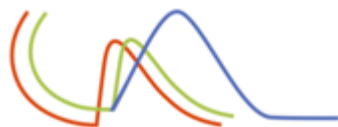
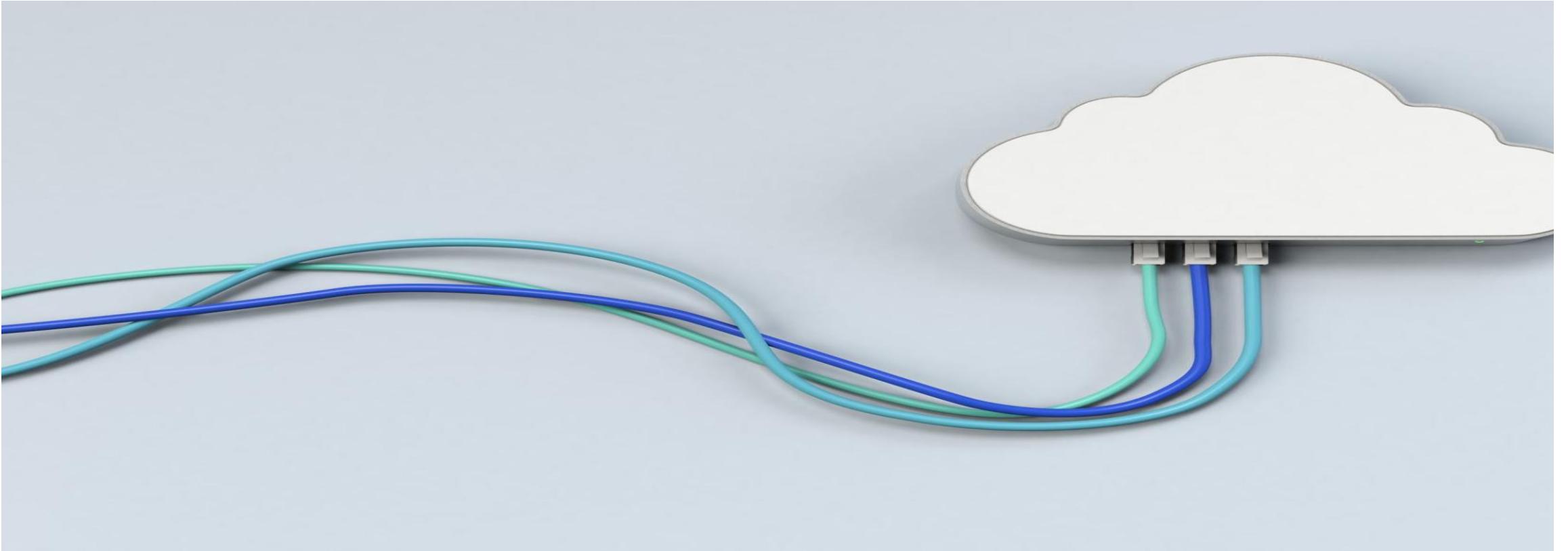
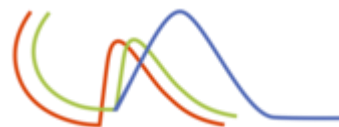
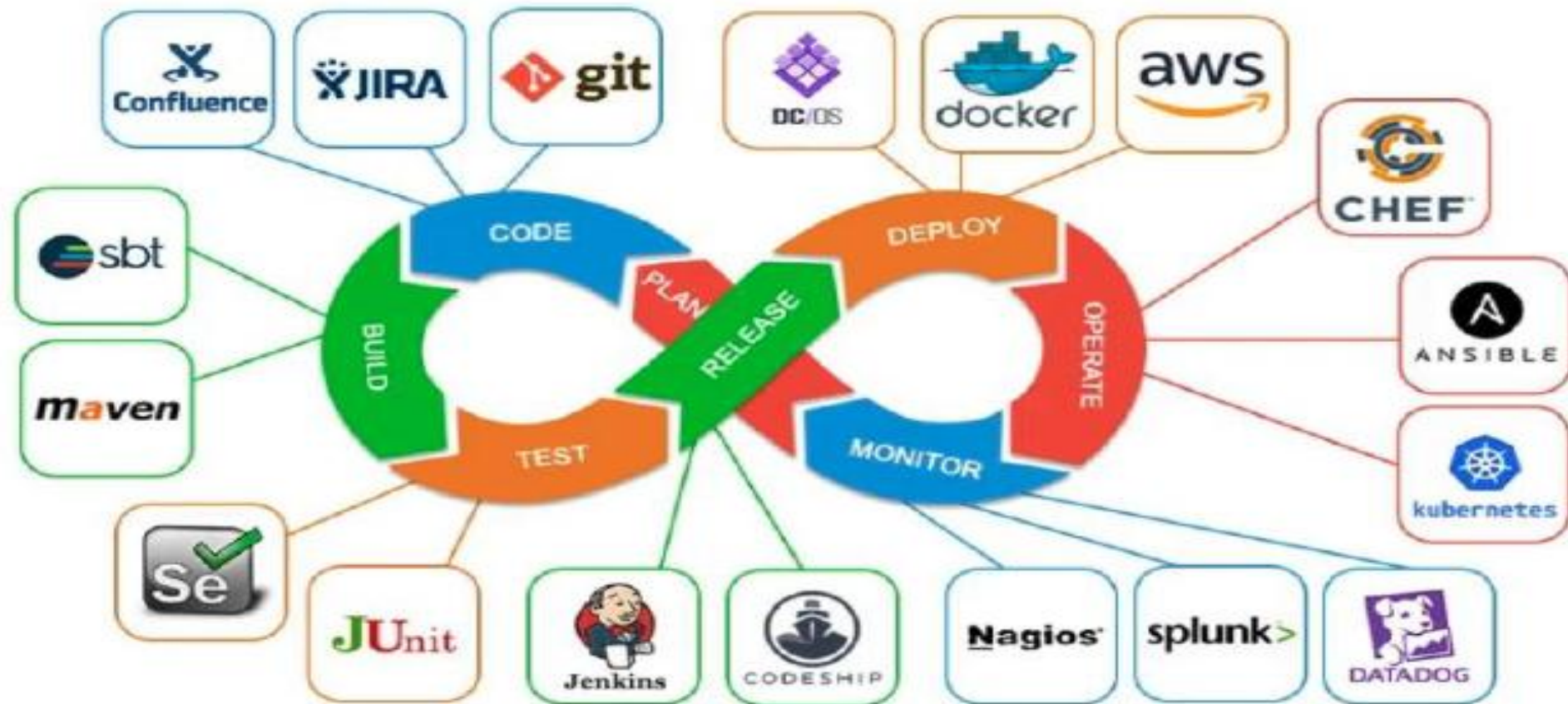


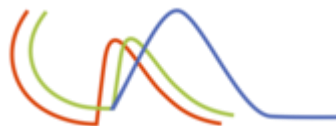
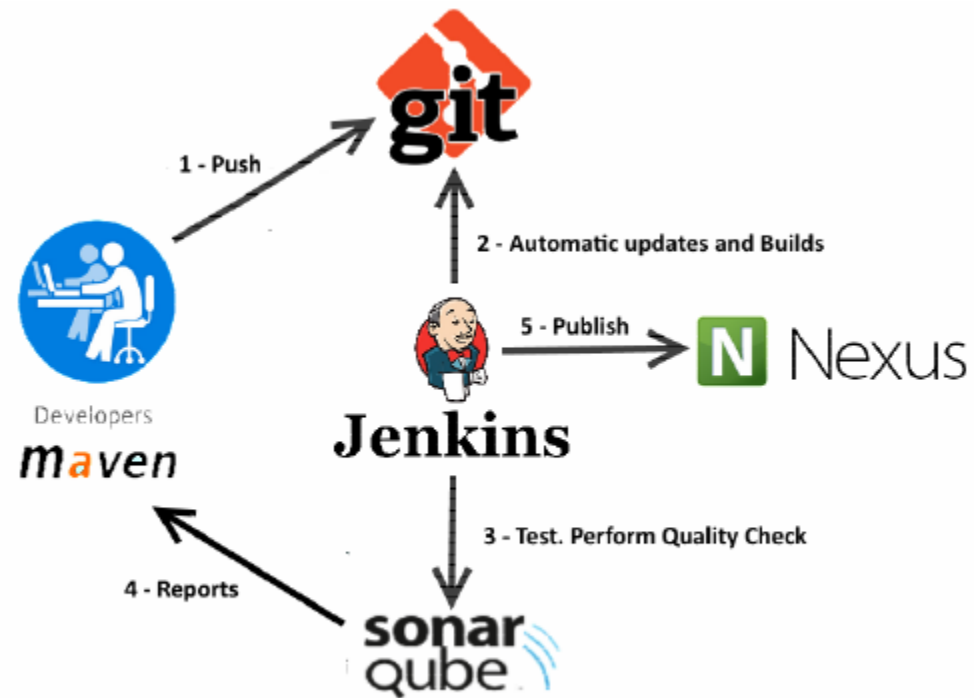
# INGENIERÍA DE SOFTWARE NATIVO PARA LA NUBE



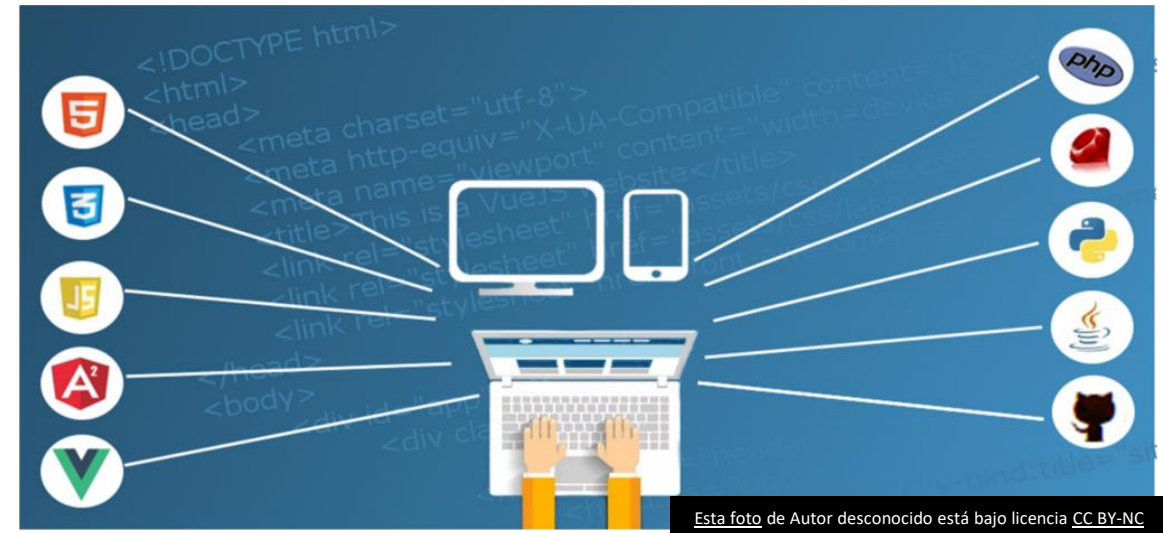
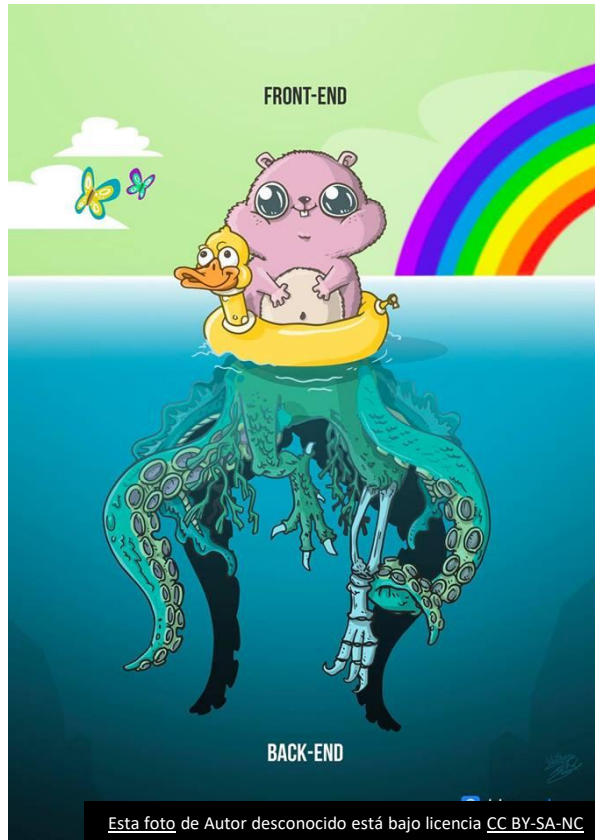
# DevOps



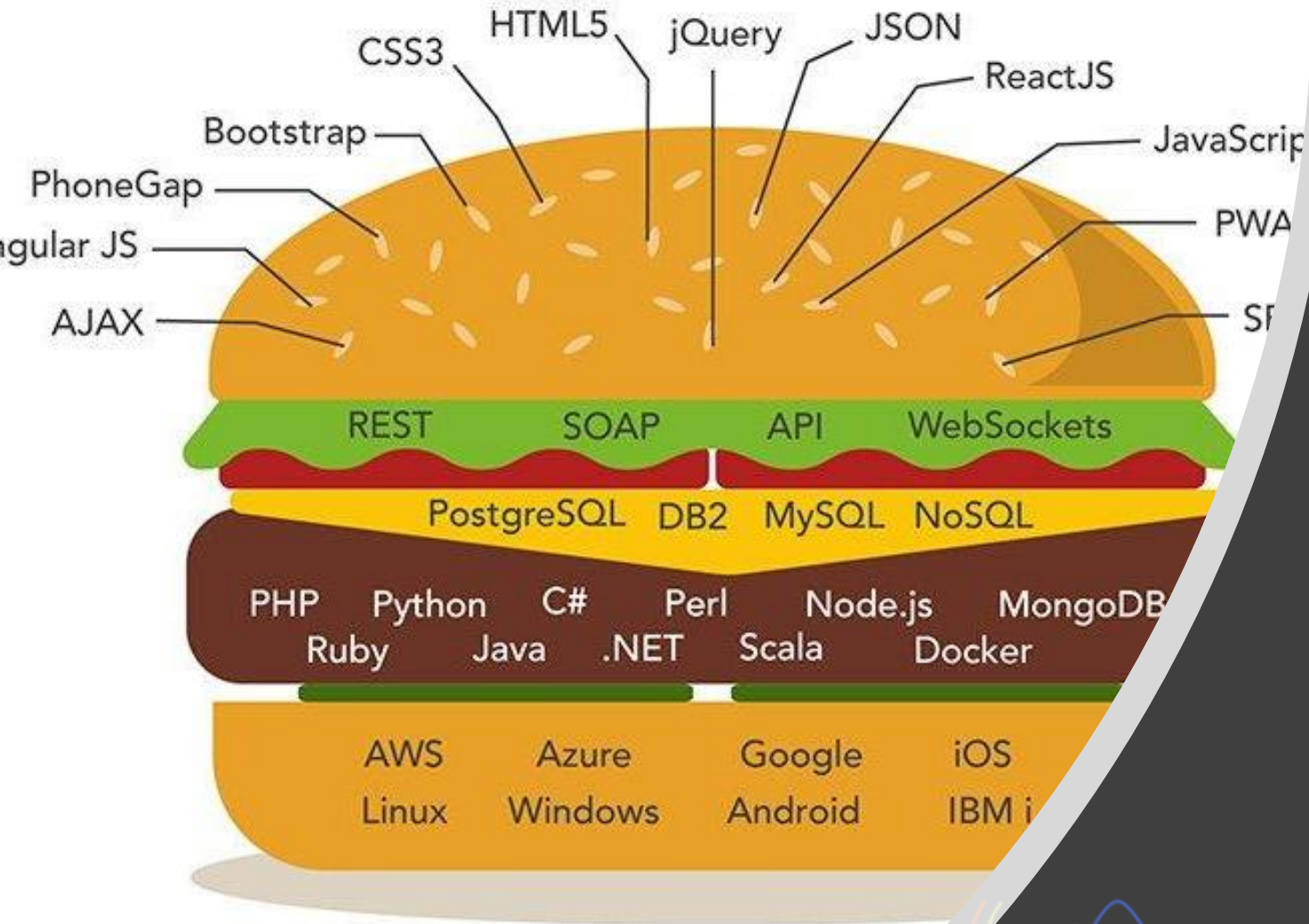
# Dev - Desarrollo



# Desarrollo front-end vs Back-end



# "The Full-Stack"



## Java Full Stack Developer

Gabriela Gómez Puente  
2022





# Patrones de diseño

---

- ¿Qué son ?
  - Representan una solución a un problema de diseño conocido.
  - Estas soluciones son planteadas en base a la experiencia adquirida por programadores expertos durante la solución del problema
- ¿ Para qué?
  - Dar respuesta a problemas comunes que se presentan durante el desarrollo de software.

# Clasificación de los patrones de diseño



## Creacionales

- Se centran en las formas de crear instancias de objetos
- Su objetivo: abstraer el proceso de instanciación y ocultar los detalles sobre cómo los objetos son creados e inicializados



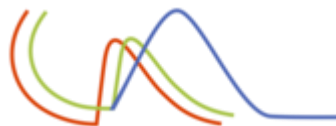
## Estructurales

- Clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades.
- Pueden ser objetos simples o compuestos



## De comportamiento

- Definen la comunicación e iteración entre los objetos de un sistema, reduciendo el acoplamiento entre ellos



# ¿Por qué se utiliza Spring?



J2EE

- EJB – Dolor de cabeza

1999–2003

JEE

- EJB

2005–2017

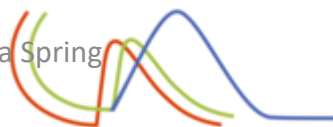
2004–2019

Rod Johnson

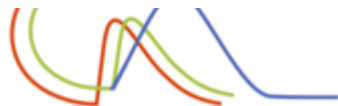




Rod Johnson



# Spring...

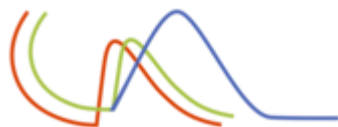


# ¿Cómo va ser el proyecto del curso?

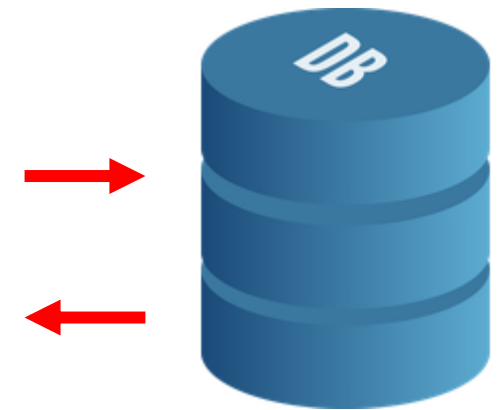
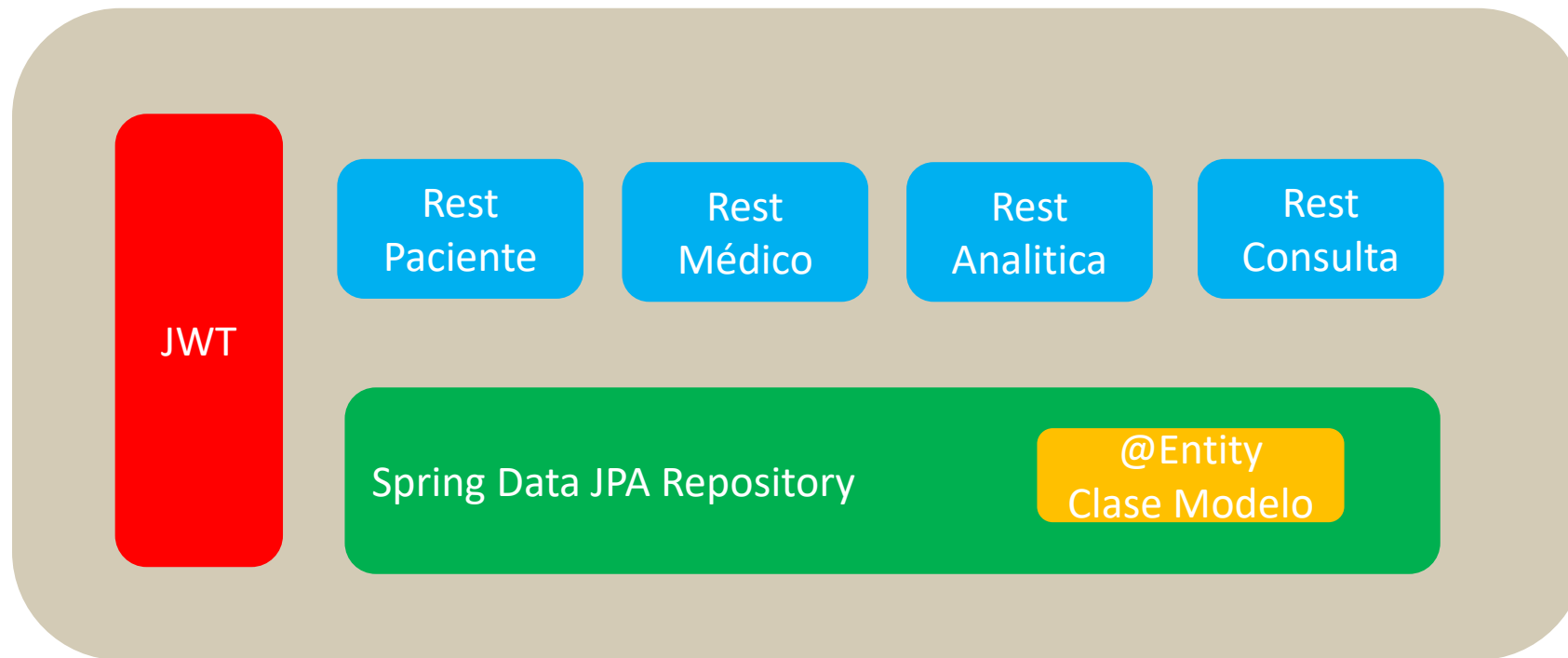


# Objetivos del proyecto

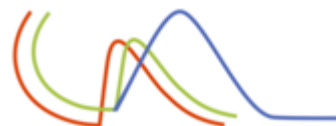
- Crear un Backend con Spring [Boot]
  - Servicios REST
  - Protección por JWT
  - Spring Data JPA Repository
  - JSON
  - Enfoque Monolítico
  - Enfoque Microservicios
- Crear un frontend con Angular
  - Angular Material
  - Seguridad y comunicación con servicios
- Puesta en producción



# Back-end [Monolito]



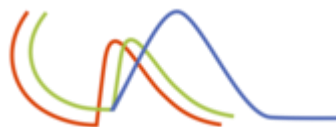
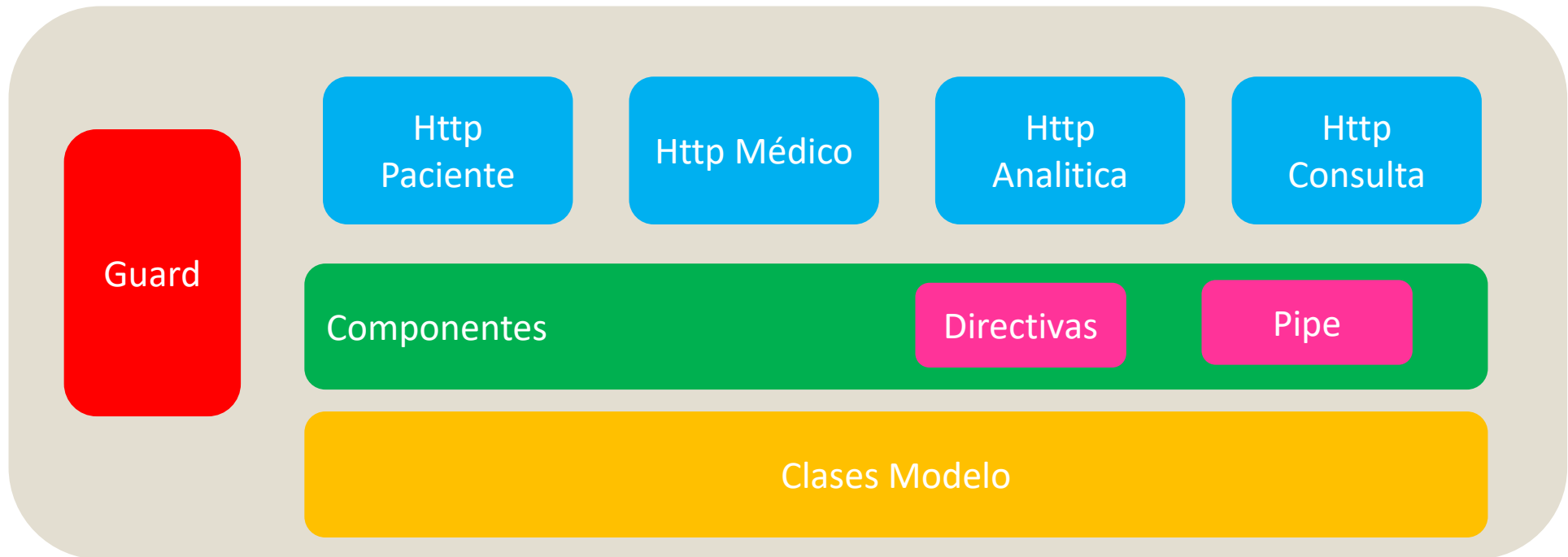
Esta foto de Autor desconocido está bajo licencia [CC BY-NC](#)



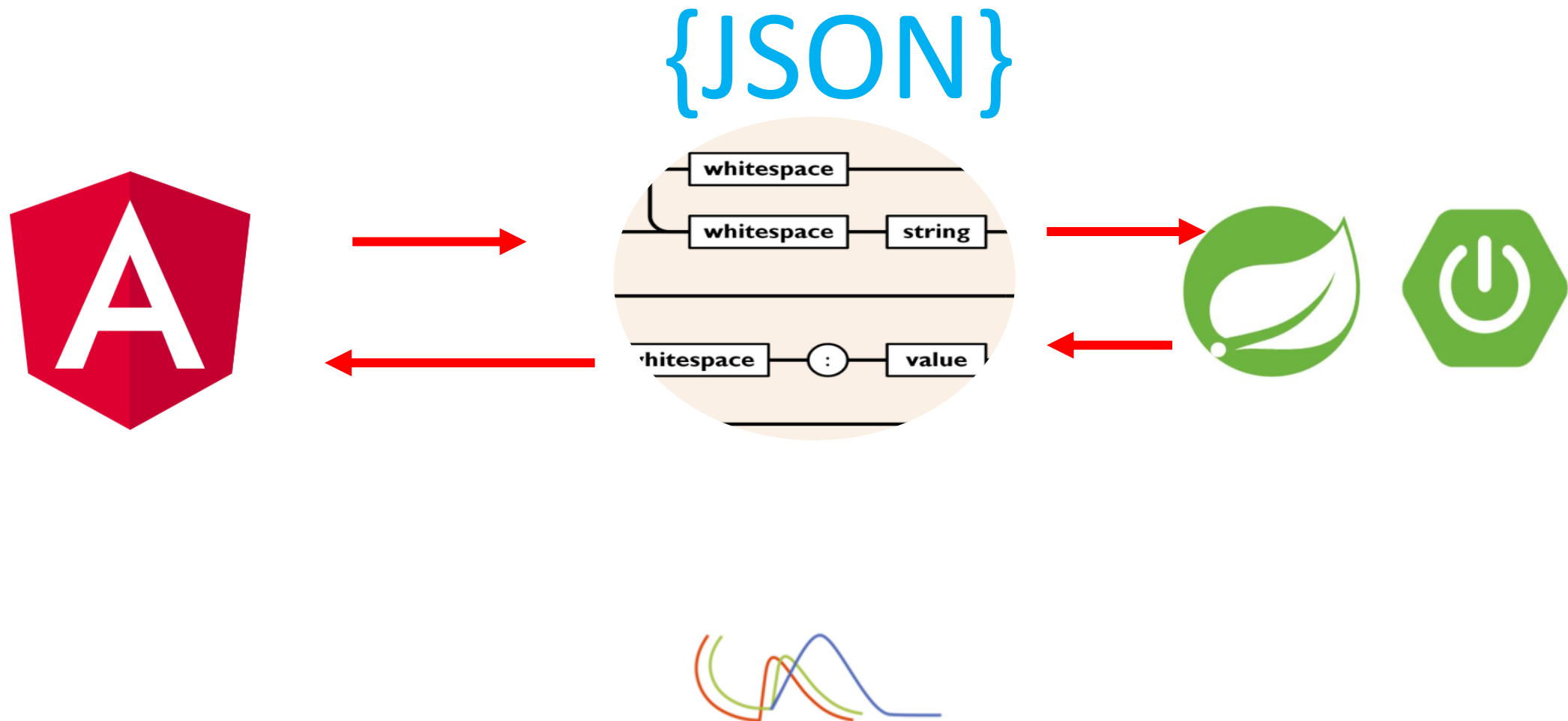


# Front-end

- Angular



# Comunicación entre ellos



- **Programación orientada a objetos**

- Trata con objetos, atributos y relaciones



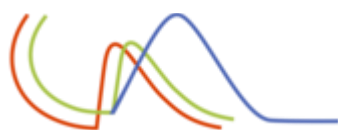
- **Uso de bases de datos relacionales**

- Trata con relaciones, tuplas y conjuntos

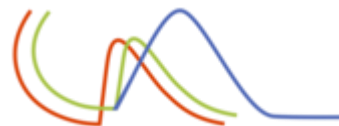
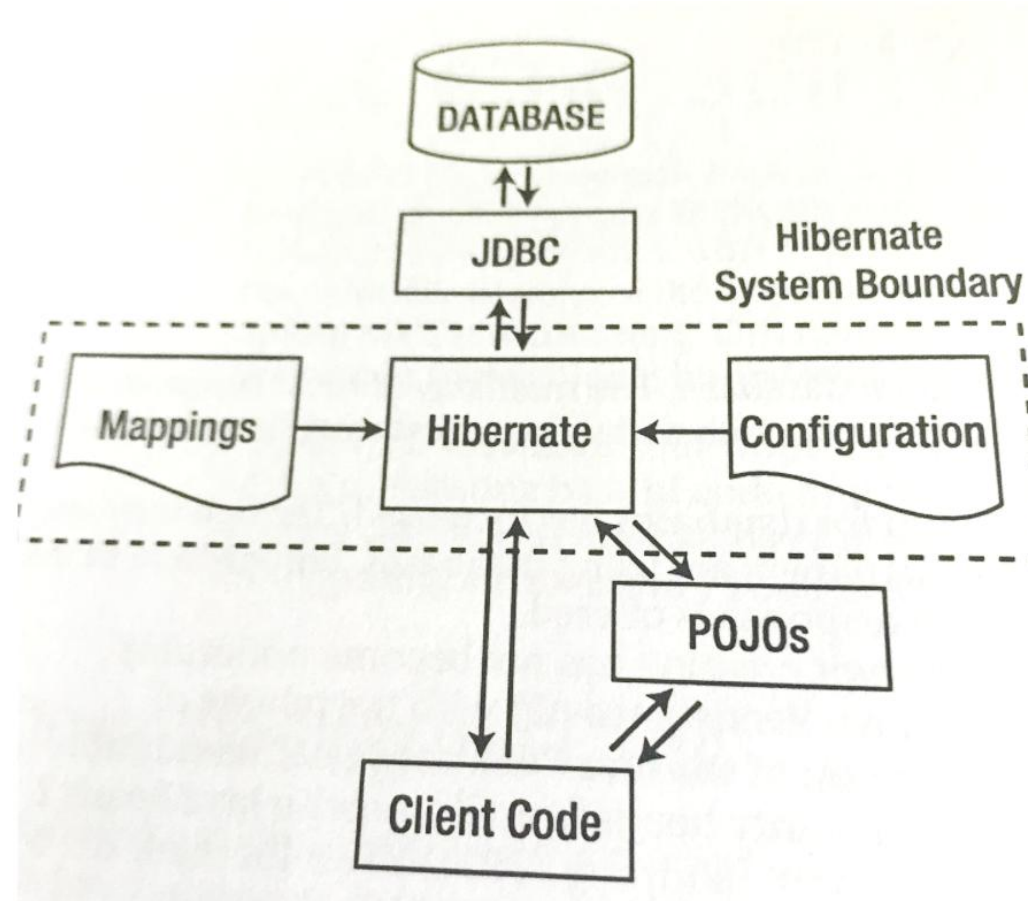


ORM: Object-Relational Mapping

- **Problema:** un 35% del código de una aplicación para realizar la correspondencia  $O \leftrightarrow R$
- **Solución:** utilizar una ORM, por ejemplo Hibernate



# Hibernate



# JPA

- JPA es el acrónimo de Java Persistence API y se podría considerar como el estándar de los frameworks de persistencia.

