

Desarrollo Back-end

ING. ALEX HERNÁNDEZ

¿Qué es JaCoCo?

- ▶ es un conjunto de herramientas de código abierto para medir y reportar la cobertura del código Java . JaCoCo se distribuye bajo los términos de la Licencia Pública de Eclipse . Fue desarrollado como reemplazo de EMMA, bajo el paraguas del complemento EcEmma para Eclipse



Características

- ▶ JaCoCo ofrece cobertura de instrucciones, líneas y sucursales.
- ▶ A diferencia de [Atlassian Clover](#) y [OpenClover](#) , que requieren instrumentar el código fuente, JaCoCo puede instrumentar el código de bytes de Java utilizando dos enfoques diferentes:
- ▶ como [JCov](#) sobre la marcha mientras ejecuta el código con un agente Java [\[2\]](#)
- ▶ como [Cobertura](#) y [JCov](#) antes de la ejecución (fuera de línea)

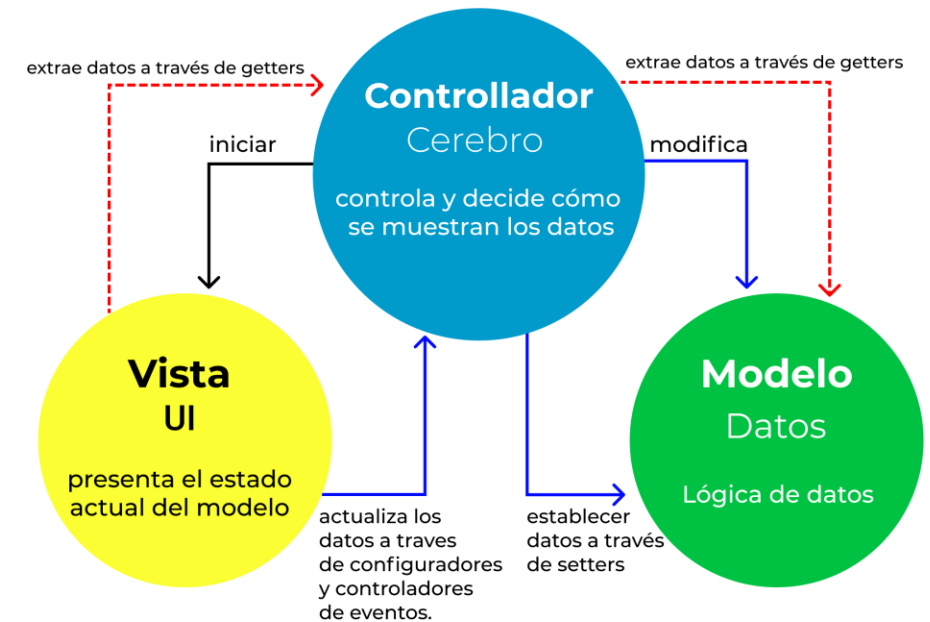
Herramientas que usan o incluyen a JaCoCo

- ▶ Complemento SonarQube JaCoCo: uno de los valores predeterminados para los análisis de cobertura dentro de la plataforma de gestión de calidad del código [SonarQube](#)
- ▶ EcEmma [Eclipse \(software\)](#) Complemento de cobertura de código, anteriormente estaba basado en EMMA ^[5]
- ▶ [Complemento Jenkins](#) JaCoCo
- ▶ [Soporte de Netbeans](#) JaCoCo
- ▶ [IntelliJ IDEA](#) desde v11
- ▶ [Complemento Gradle](#) JaCoCo
- ▶ [Servicios de equipo de Visual Studio](#)
- ▶ [Ciudad del equipo](#) ^[10]
- ▶ SELLO (<https://www.stamp-project.eu/>)

MVC (Modelo Vista Controlador)

- es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Patrones de Arquitectura MVC



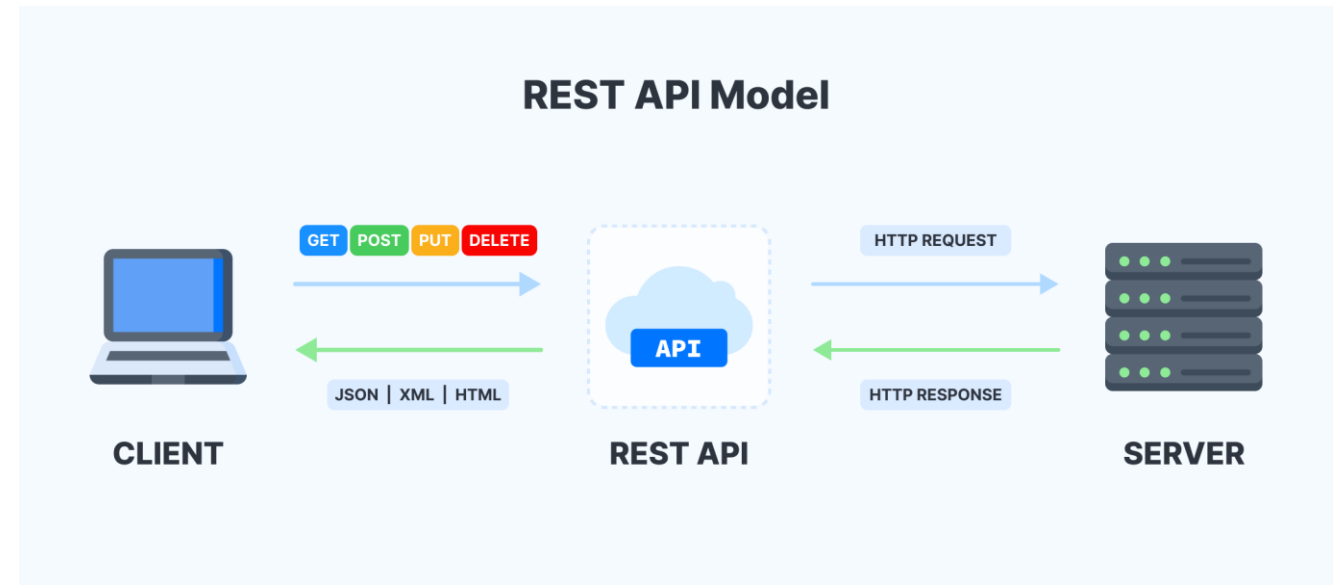
Descripción del patrón

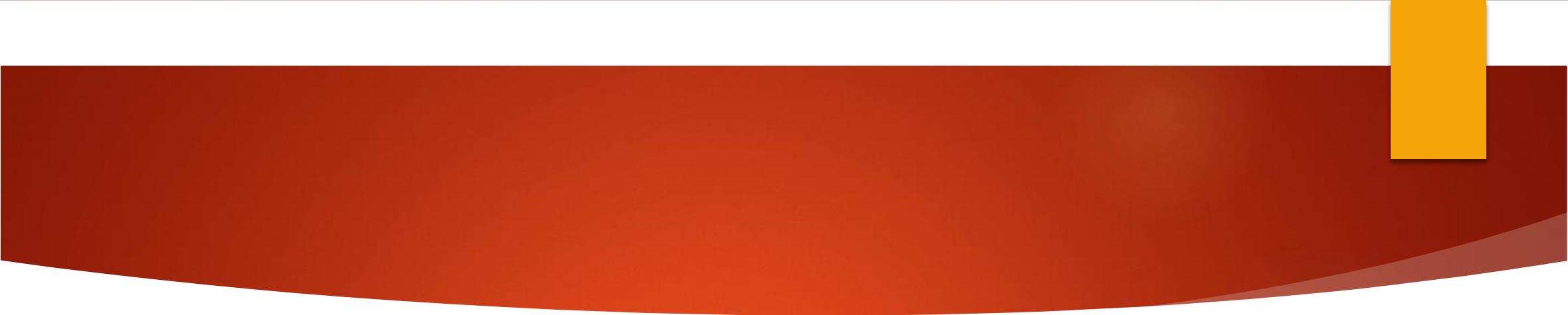
De manera genérica, los componentes de MVC se podrían definir como sigue:

- ▶ El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación ([lógica de negocio](#)). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.¹²
- ▶ El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (véase [Middleware](#)).
- ▶ La **Vista**: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la [interfaz de usuario](#)), por tanto requiere de dicho 'modelo' la información que debe representar como salida.

¿Qué es una API de REST?

- ▶ Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).



- 
- ▶ Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta). Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima.
 - ▶ En otras palabras, las API le permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla.