

## DESCRIPCIÓN DEL PROYECTO

-El programa recibe eventos de pedidos que se pasan por **post** usando la aplicación **Postman** al controlador *PedidoController*, ya sea con formato **json** o **xml**.

-El controller los recibe como una lista de pedidos y los envía al *PedidoService* para ser tratados.

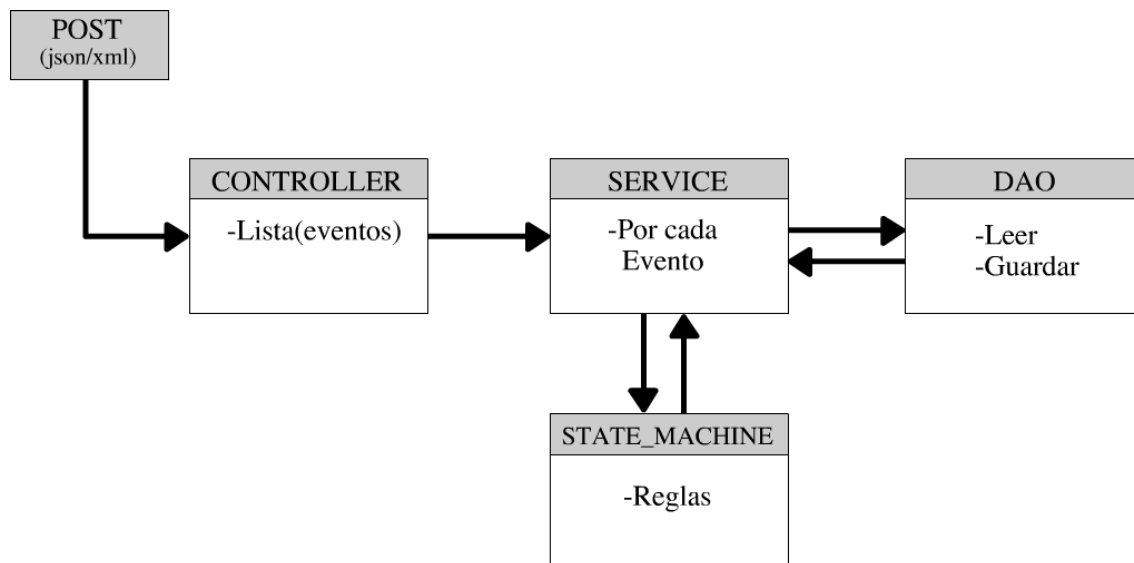
-Una vez tratado cada evento, se mira si el pedido está ya guardado previamente en la base de datos y de ser así se recoge el último estado en el que se encuentra.

-Se comprueba si, de existir en el sistema, el pedido puede cambiar al estado recibido, y si es así se guarda el pedido con el nuevo estado.

Los eventos se guardan en la base de datos siempre que el pedido con id *order\_id* no haya sido guardado previamente en la base de datos y también si, estando presente en la base de datos, el estado resultante tras evaluar si puede o no cambiar de estado, es diferente al que tenía previamente y no es el estado final.

Un estado puede pasar a un estado diferente, siempre que el estado al que se quiere transitar no sea el estado inicial, siempre que el estado previo al intento de transición no sea el estado final y siempre que exista el estado al que se quiere transitar.

Los estados y eventos están definidos como *Enum*.



En un principio se intentó hacer el ejercicio completo usando **Spring StateMachine**, pero al final, si bien se usa la máquina de estados para transitar de un estado a otro, no cambiando de estado si la transición no está permitida (según las reglas que se especifican en la configuración de la máquina), sacando el estado al que se ha transitado por consola a través del listener definido en la configuración, se optó por controlar la transición e inserción de los datos de forma manual, si bien el cambio de estado se realiza usando **Spring StateMachine**.

El proyecto ha sido realizado usando IDE IntelliJ IDEA Community, Spring Boot, Spring StateMachine, Jackson y BD H2.