

# Callbacks(hooks)



**Callbacks de suscripción y solicitud** Estos se ejecutan cuando el Subscriber interactúa con el Publisher.

Callback	Descripción
<code>doOnSubscribe(Consumer)</code>	Se ejecuta cuando un Subscriber se suscribe al flujo. Útil para logs o inicialización.
<code>doOnRequest(LongConsumer)</code>	Se llama cuando el Subscriber solicita elementos ( <i>request(n)</i> , <i>backpressure</i> ).
<code>doOnCancel(Runnable)</code>	Se ejecuta si el flujo se cancela antes de completarse. Puede ser por <i>subscription.cancel()</i> , <i>dispose()</i> , o por un operador como <i>take(n)</i> .

**Callbacks de procesamiento de elementos** Estos se activan cuando los datos fluyen a través del Flux o Mono.

Callback	Descripción
<code>doOnNext(Consumer)</code>	Se ejecuta cada vez que un nuevo elemento es emitido por el Publisher. Útil para depuración o logs.
<code>doOnEach(Consumer&lt;Signal&gt;)</code>	Se ejecuta en cada señal emitida ( <i>onNext</i> , <i>onError</i> , <i>onComplete</i> ). Permite inspeccionar la señal completa.

**Callbacks de finalización** Estos se disparan cuando el flujo termina por éxito o error.

Callback	Descripción
<code>doOnComplete(Runnable)</code>	Se ejecuta cuando el flujo finaliza con éxito.
<code>doOnError(Consumer)</code>	Se ejecuta si ocurre un error.
<code>doOnError(Class exceptionType, Consumer)</code>	Se ejecuta si ocurre un error de un tipo específico.
<code>doOnTerminate(Runnable)</code>	Se ejecuta cuando el flujo termina, sea por éxito o error. Es similar a <i>doOnComplete</i> + <i>doOnError</i> .
<code>doFinally(Consumer)</code>	Se ejecuta al final del flujo, indicando la razón ( <i>CANCEL</i> , <i>COMPLETE</i> , <i>ERROR</i> ).

**Callbacks globales (Hooks)** Son métodos estáticos para configurar comportamientos en toda la aplicación Reactor.

Callback	Descripción
<code>Hooks.onOperatorDebug()</code>	Habilita <i>stacktraces</i> detallados para depuración.
<code>Hooks.onEachOperator(Function&lt;Publisher, Publisher&gt;)</code>	Permite modificar cada operador aplicado a un flujo.
<code>Hooks.onLastOperator(Function&lt;Publisher, Publisher&gt;)</code>	Modifica el último operador en la cadena del flujo.
<code>Hooks.onNextDropped(Consumer)</code>	Maneja elementos descartados sin consumir.
<code>Hooks.onErrorDropped(Consumer)</code>	Maneja errores que se ignoran sin manejar explícitamente.