Event Sourcing

Registra todos los cambios al estado de una aplicación como una secuencia de eventos.

Índice

¿De qué vamos a hablar?

1. Definiciones. *(¿qué es y cómo funciona?)*
2. Posibles aplicaciones.
3. Ventajas. *(¿qué ganamos?)*
4. Casos de éxito. *(¿quién lo está usando?)*
5. Inconvenientes. *(¿de verdad merece la pena?)*
6. Futuro.

1

¿Qué?

¿Cómo?

¿Sería posible el desarrollo del software en la actualidad sin sistemas de control de versiones?

NO.

Estaríamos obligados a mantener un ***registro manual*** de cambios en los propios ficheros.

```
# Change:     14532
# Update At: 2018-04-23
# Update By: Philippe Creux
# Reason:     Fix a bug that was introduced sometimes between
#             change #12320 that's in users.rb.backup-2018-02-11
#             and change #14211 above (around line 2400)


# def destroy
#   self.destroy_at = Time.now
# end


def destroy
  self.deleted_at = Time.now
end
```

Problema

Alguien introduce un fallo de seguridad en un componente poco utilizado.

Al cabo de un tiempo, la funcionalidad que emplea el componente cobra relevancia.

Se detecta el problema de seguridad pero el origen del problema es desconocido.

¿Solución?

Analizar los registros de cambios en miles de ficheros para buscar el que ha producido el fallo.

¿Solu ?

Analizar gistros bios en s de ficheros uscar el q rod el fallo.

Solución

Emplear un sistema de control de versiones que registre todos los **cambios en el estado** del código y nos permita devolverlo a cualquiera de ellos.

Ya tenemos el código bajo control, pero...

¿Qué pasa con los datos de la aplicación?

Event Sourcing

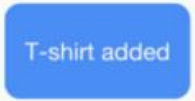
Registra de manera centralizada todos los cambios en el estado de una aplicación junto con el origen y el momento en el que ocurrieron.

GIT, para datos

Event Sourcing

Registra de manera centralizada **todos los cambios en el estado de una aplicación** junto con el origen y el momento en el que ocurrieron.

Una tienda de ropa



Algunas definiciones (I)

Las acciones del usuario, llamadas a APIs o métodos, trabajos cron… generan **EVENTOS**.

Nos interesa conocer:

- **Cuando** han ocurrido.
- **Quien** los ha generado.
- **Cuál** era su estado anterior.

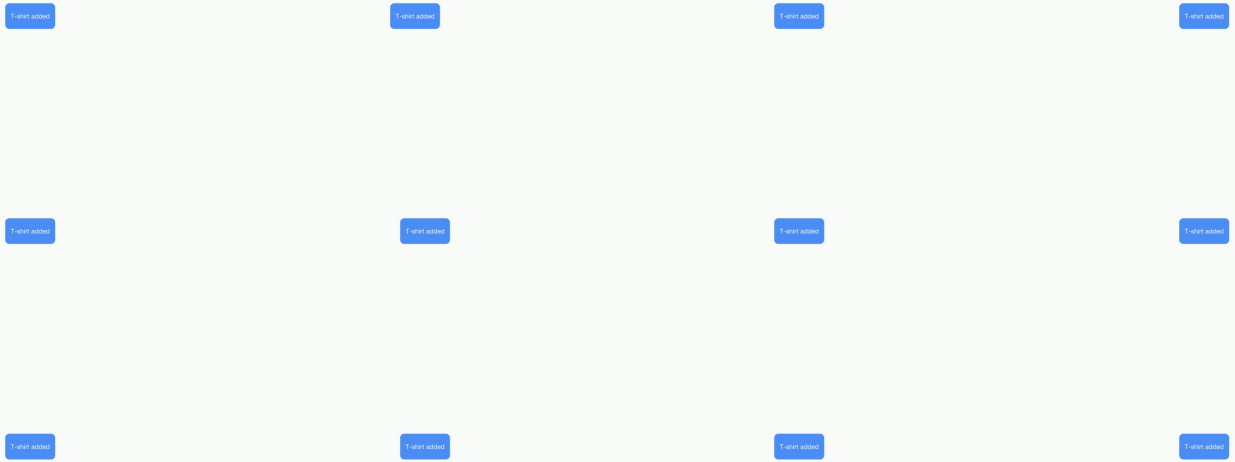
Algunas definiciones (II)

Los eventos se registran de forma centralizada en un **REGISTRO.**

Esto nos permite:

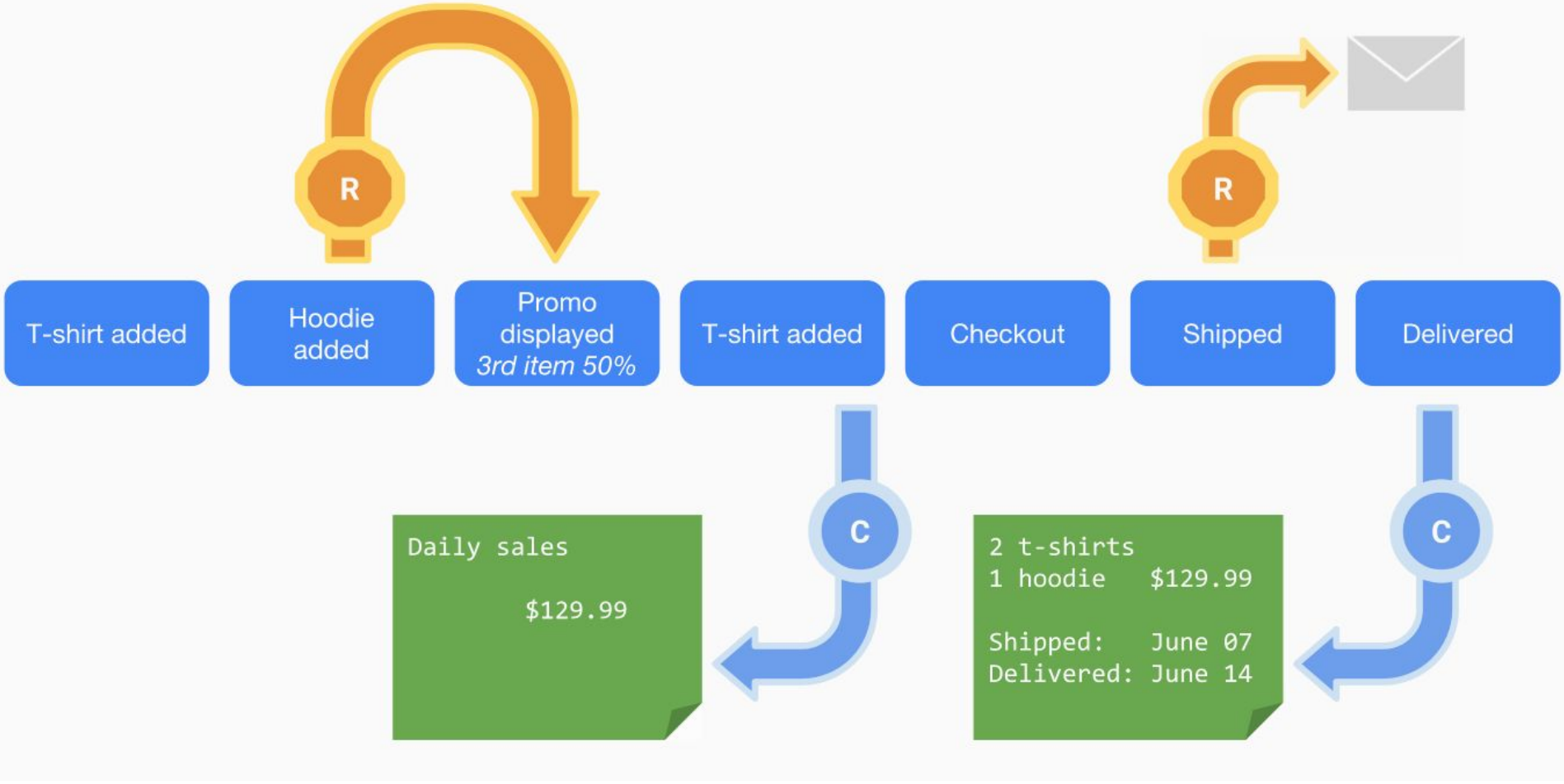
- **Reconstruir** el estado de la aplicación repitiendo los eventos del registro.
- **Repetir** secuencias concretas de eventos.
- **Consultar** cual era su estado anterior.

50 tiendas de ropa



3 elementos básicos

1.
2. **Agregadores** representan de manera resumida (*agregada*) el estado actual de la aplicación.
3. **Calculadores** leen una secuencia de eventos y los modifican como sea necesario..
4. **Reactores** *reaccionan* a los eventos a medida que ocurren.


R
R
T-shirt added
Hoodie added
Promo displayed
3rd item 50%
T-shirt added
Checkout
Shipped
Delivered
C
C
Daily sales
$129.99
2 t-shirts
1 hoodie $129.99
Shipped: June 07
Delivered: June 14

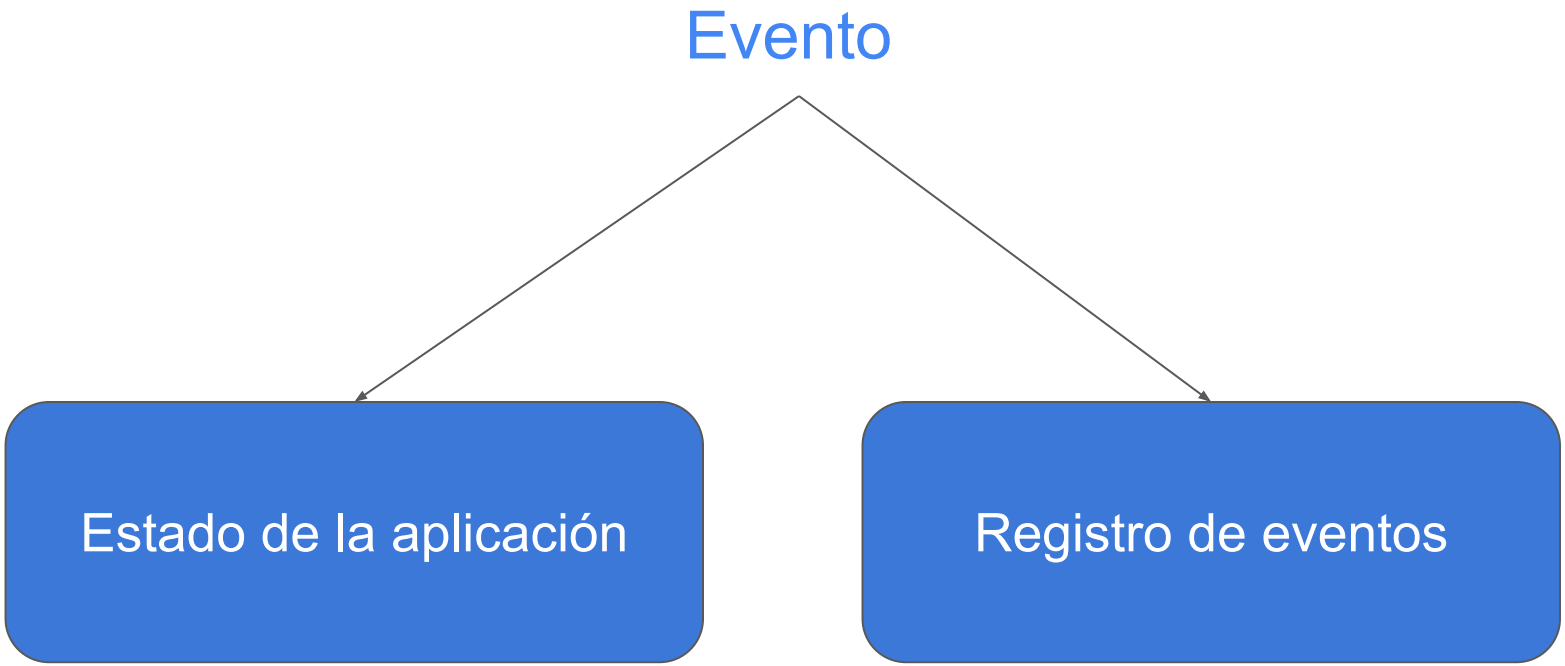
2 Aplicaciones

Cuando una **tarea implica varios pasos** y en caso de error, para revertir los cambios sería necesario reproducirlos de nuevo en orden para devolver los datos a un estado coherente

Cuando es necesario **capturar la intención, el propósito o el motivo** en los datos. Por ejemplo, los cambios en una entidad de cliente se pueden capturar como una serie de tipos de evento específicos como Cambio de domicilio, Cierre de cuenta o Fallecimiento.

Cuando el **uso de eventos** es una **característica natural** de la operación de la aplicación y requiere poco esfuerzo de implementación o desarrollo adicional

3 ¿Qué ganamos?


Evento
Estado de la aplicación
Registro de eventos

Gracias al registro de eventos:

Reconstrucción completa

Consulta temporal

Reproducción de eventos

Ventajas:

- Desacoplamiento (productor - consumidor)
- Trazabilidad
- Escalabilidad
- Tolerancia a fallos / Reconstrucción

- A partir registros -> bases de datos.
- Instantáneas en cualquier momento.
- Agregar fácilmente funciones al sistema.
- Reacción rápida.
- Eventos independientes de los detalles técnicos del sistema.
- Auditar o reproducir el historial.

4 ¿Quién lo está usando?

Sistemas de control de versiones:

Sistemas de bases de datos:

Sistemas contables:

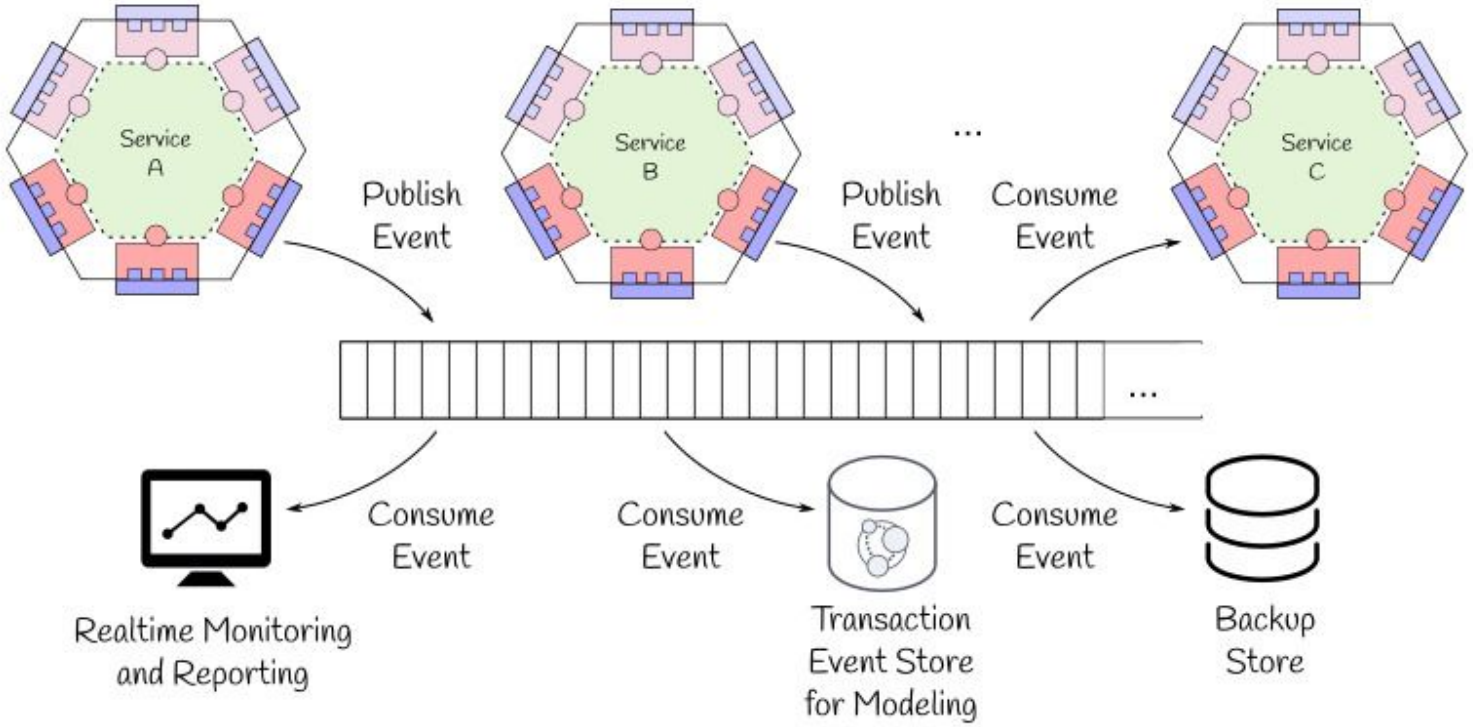
	A	B	C	D	E
1				**Saldo:**	1.304,45 €
2					
3	**Fecha**	**Concepto**	**Entrada**	**Salida**	**Saldo**
4	7/9/18	Movimiento inicial	100,00 €		100,00 €
5	7/9/18	Ingreso general	1.000,00 €		1.100,00 €
6	8/9/18	Comida		56,00 €	1.044,00 €
7	8/9/18	Gasolina		40,00 €	1.004,00 €
8	9/9/18	Intereses	0,45 €		1.004,45 €
9	9/9/18	Traspaso de otras cuentas	300,00 €		1.304,45 €

5 ¿Es una buena idea?

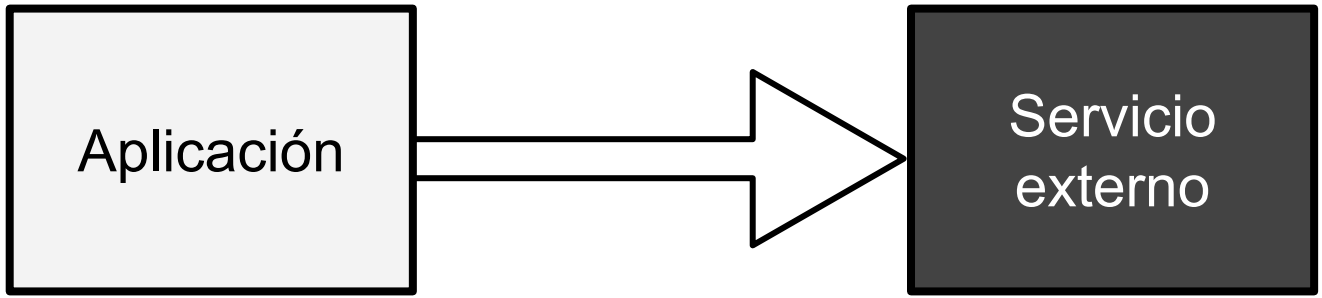
Mucho más código

- **Generar** eventos
- **Guardar** eventos
- **Transiciones / Procesamiento** de eventos

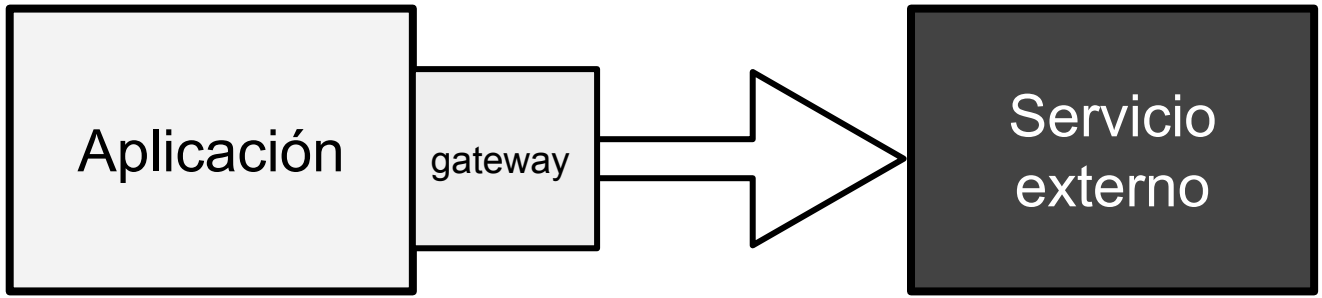
Acoplamiento a los eventos



Interacción exterior

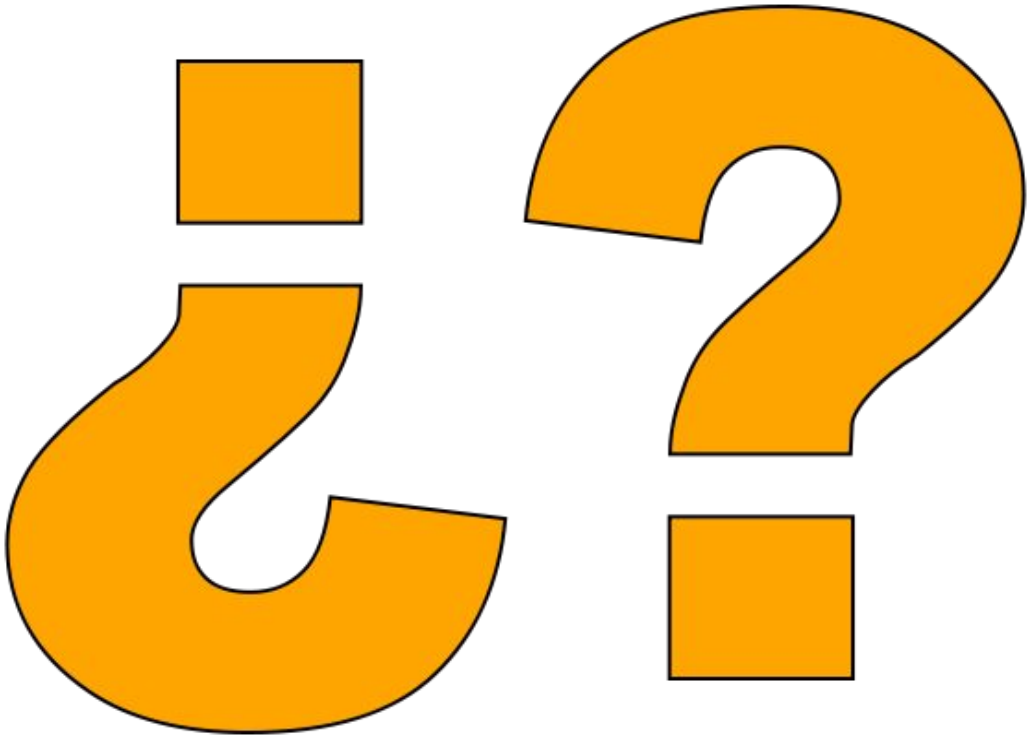


Interacción exterior



¿Se quedará obsoleto?


Jest
Redux
jQuery
Bootstrap
NEXT.JS
Flask
spring
Framework
node
JS
RAILS
django
Express JS
.NET
Core



Bibliografía

https://docs.microsoft.com/es-es/azure/architecture/patterns/event-sourcing

https://kickstarter.engineering/event-sourcing-made-simple-4a2625113224

https://martinfowler.com/eaaDev/EventSourcing.html

https://microservices.io/patterns/data/event-sourcing.html