

Armando J. R. Nunes

# BRAINSTORMING SYSTEM



Usando o Event Storming para  
conhecer um domínio de  
negócio.

*A minha interpretação*

Petrópolis, Rio de Janeiro

# EVENT STORMING

KNOWLEDGE CRUNCHING

## O que é o Event Storming

Event Storming ([https://en.wikipedia.org/wiki/Event\\_storming](https://en.wikipedia.org/wiki/Event_storming)) é um método de brainstorming em que o conhecimento e a compreensão de um campo de conhecimento específico e delimitado (um domínio de negócios) são desenvolvidos e visualizados conjuntamente num workshop. O ponto de partida são os chamados eventos de domínio.

"É um formato de workshop para explorar domínios complexos"

- Alberto Brandolini



## Informações Gerais

(Configuração para sessões distribuídas de Event Storming)

Pessoas  
Entre 4 e 10

Tempo de  
Duração  
Entre 2 e 6  
Horas

Tempo de  
Preparação  
20 minutos

Dificuldade  
Alta

## Quem deve estar envolvido?

É importante para um workshop de Event Storming ter as pessoas certas presentes. Isso inclui pessoas que sabem as perguntas a serem feitas e aquelas que sabem as respostas.

- 1) Pessoas que sabem as perguntas - Arquitetos, Desenvolvedores e Testadores
- 2) Pessoas que sabem as respostas - Especialistas de Domínio, Product Owners, Clientes
- 3) Um bom moderador e facilitador - Scrum Master, Agile Coach, Arquiteto

## Preparação

Certifique-se sobre a área de conhecimento que você deseja modelar ou entender.

Quanto mais extenso o processo de negócios a modelar, mais tempo este processo levará.

Pense na possibilidade de dividir a área de conhecimento em processos de negócios a modelar em processos menores, e na possibilidade de modelar estes sub processos em separado e um após o outro.

Por exemplo:

- Área de Conhecimento: Loja Online
- Processos de Negócios: Ofertas da Loja, Processo de Pedidos, Processo de Distribuição, Processo de Recebimento, etc.

## Terminologia

### Domain Event

#### Evento de Domínio

Captura a memória de fatos que ocorreram como estímulo ao domínio. Deve ser escrito sempre no passado.

### Command

#### Comando

Representa uma ação, interação ou decisão que leva ao evento ao qual está relacionado. Deve ser escrito como verbo na forma imperativa.

### Actor

#### Usuário/Ator

Uma pessoa no desempenho de um papel que interage com o domínio e toma decisões.

### Process / Policy

#### Política / Processo de Negócio

Representa as regras de negócio e as lógicas que regem os eventos. Indica uma decisão a ser tomada.

### External System

#### Sistema Externo

Representa um sistema externo mantido por outro time ou organização fora da influência do domínio.

### Aggregate

#### Agregado

Objeto que vai processar o comando. É onde acontece o Evento. Atua como máquina de estado que garante a consistência do negócio. Deve ser definido como um substantivo. É um cluster de conceitos correlacionados.

### Read Model

#### Visão / Modelo de Leitura

É uma visão com a qual os Atores interagem para que possam tomar uma decisão ou executar uma tarefa no domínio.

### UI

#### Interface de Usuário

Representa uma interface de usuário na qual uma Visão será retornada ou exibida a um Ator.

### Hot Spot

#### Ponto de Interrogação / Risco

Representa dúvida, risco para o negócio ou ponto obscuro que será posteriormente esclarecido.



Evento disparado a partir de um momento no tempo.



Resultado de exceção de uma Política de Negócio.



Resultado normal ou esperado de uma Política de Negócio.



# BIG PICTURE

Obter a visão do fluxo de eventos

## Etapa 1 - Descoberta de Eventos (Big Picture)

*Tempo*

De 30 a 45  
Minutos

**Objetivo:** Coletar os Eventos de Domínio.

### Quais eventos existem na sequencia do fluxo do processo?

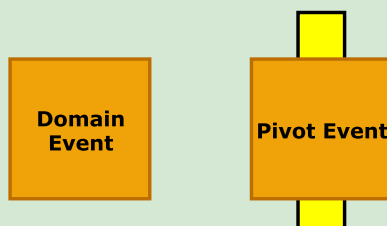
Um Evento de Domínio é qualquer evento que seja importante para o negócio e que tenha um impacto explícito nele, podendo acontecer dentro ou fora dele.

Por convenção, cada evento é expresso através de um verbo no passado.

O facilitador geralmente dá o primeiro passo fixando eventos cruciais chamados Pivot Events (através de barras verticais amarelas que cobrem toda a altura do board) que enfatizam os principais acontecimentos do negócio de ponta a ponta.

A primeira rodada é um processo de brainstorming puro, uma exploração caótica.

**Post-its usados nesta etapa:**



## Etapa 2 - Refinamento Temporal (Big Picture)

*Tempo*

De 30 a 60  
Minutos

**Objetivo:** Refinar os Eventos de Domínio.

### Qual a ordem que os eventos deveriam estar e quais eventos deveriam existir e ainda não estão no board ?

Nesta fase percorremos os post-its com os participantes para entendimento e validação.

Quando surgirem dúvidas, pedimos aos participantes que expliquem o que cada evento significa.

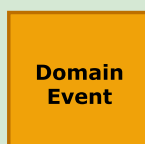
Devemos também verificar se os eventos estão na ordem certa em termos de tempo.

Devemos unificar os sinônimos (termos diferentes que significam a mesma coisa) e aprimorar as diferenças se o mesmo termo for usado para representar coisas diferentes.

O foco aqui não é agrupar e sim colocar todos os post-its em ordem temporal da esquerda para a direita.

Se ficar complicado organizar os subprocessos, devemos usar as separações verticais, mas devemos manter o foco em colocar os eventos em ordem, não os grupos.

**Post-its usados nesta etapa:**



# EVENT STORMING

KNOWLEDGE CRUNCHING

## PROCESS MODELING

Explorar processos de negócios, modelar épicos ou features

### Etapa 3 - Detecção de Gatilhos (Process Modeling)

Tempo

De 1 a 3  
Horas

**Objetivo:** Identificar as causas dos Eventos de Domínio.

#### Quais as causas dos nossos eventos?

Aqui devem ser levantados os fatores que geram os eventos (Comandos, Usuários/Atores, Visões/Modelos de Leitura,

Sistemas Externos, tempo (por exemplo eventos temporais), Processos de Negócio e outros Eventos de Domínio (através de reações automáticas).

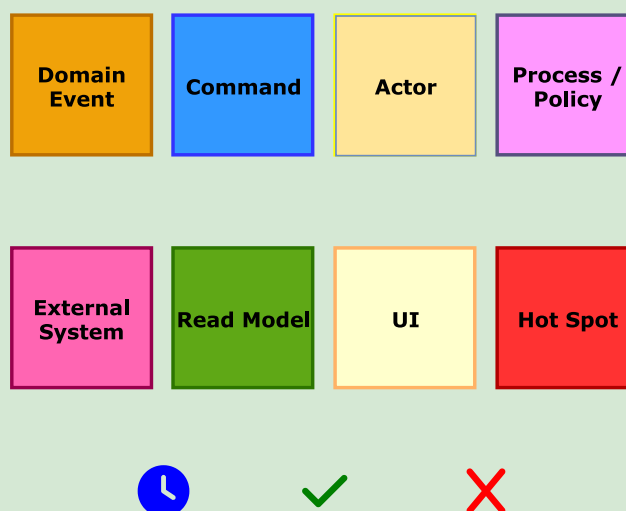
Alguns eventos são a consequência direta de uma ação do usuário, são representados pelos Comandos.

Outros são consequência de algo acontecendo em sistemas externos ou através da passagem do tempo.

Em alguns casos, teremos eventos que serão a consequência direta de outros eventos. Nestes casos devemos colocar os eventos próximos.

Durante este processo é comum observar que um comando irá gerar vários eventos.

#### Post-its usados nesta etapa:





# SOFTWARE DESIGN

Modelagem dos agregados, contextos e possíveis microsserviços

## Etapa 4 - Categorização de Entidades e Agregados (Software Design)

*Tempo*

De 60 a 90  
Minutos

**Objetivo:** Reclassificação e resultado da Modelagem de Software.

**Que usuários disparam quais eventos?**

**Que comandos afetam que agregados ou sistemas externos e desencadeiam que alterações?**

**Que agregados ou sistemas externos acionam que eventos durante o processamento de comandos?**

**Que eventos acionam que políticas?**

**Que eventos criam que modelos de leitura para os casos de uso?**

**Que políticas chamam que novos comandos?**

Agora é hora de procurar os substantivos nos eventos, remover os post-its da linha do tempo e agrupá-los em torno dos agregados encontrados.

Os agregados são a parte do sistema que recebe os comandos e que geram os eventos. Eles são os objetos que armazenam os dados e são modificados pelos comandos (entidades formadas por associação entre entidades e complementadas por mais algumas informações. Por exemplo: Produto, Pedido, etc).

Para esta etapa inserimos os post-its de Aggregate. Eles devem ser colocados acima e entre os pares de post-its comando/evento já mapeados e o padrão de nomenclatura é a utilização do substantivo que melhor represente o agregado.

Importante notar que em vez de definir agregados a partir do código, aqui se adota uma abordagem de fora para dentro, ou seja, o agregado é a parte do sistema que recebe os comandos e decide se os executa ou não, produzindo assim um evento de domínio.

Pode ser usado um outro recurso que são as Swimlanes. Elas separam o fluxo em raias horizontais, atribuídas a determinados atores ou departamentos, com todos os seus componentes dispostos abaixo. Elas melhoram a legibilidade e são uma escolha mais óbvia quando o grupo de pessoas e as discussões já estão em um nível mais maduro.

**Post-its usados nesta etapa:**

**Aggregate**

Departamento Comercial





# SOFTWARE DESIGN

Modelagem dos agregados, contextos e possíveis microsserviços

## Etapa 5 - Categorização de Fronteiras (Software Design)

*Tempo*

De 60 a 90  
Minutos

**Objetivo:** Reclassificação e resultado da Modelagem de Software.

### Em qual contexto os objetos estão inseridos?

Após toda a discussão realizada no workshop, possivelmente já é possível identificar os prováveis domínios do negócio. Eles podem estar relacionados a divisões departamentais (as Swimlanes ajudam nesse sentido), diferentes visões que os especialistas de negócio possuem sobre o mesmo conceito ou, até mesmo, entidades que são importantes ao núcleo principal do domínio (core domain).

As fronteiras sugerem o conceito de Contextos Delimitados que são espaços no qual uma terminologia específica pode ser compreendida de forma onipresente, geralmente envolvendo uma única equipe com poucos interessados.

Devemos agrupar os itens que fazem parte do mesmo domínio e desenhar círculos para delimitá-los. Também podemos usar um post-it de cor diferente para dar nome ao domínio.

Também podemos utilizar esta etapa para mapear eventos que "naveguem" entre domínios e o sentido em que eles "navegam". Assim é possível ter uma visão de alto nível sobre o fluxo dos processos.

### Elementos usados nesta etapa:

Nome do Domínio



# SOFTWARE DESIGN

Modelagem dos agregados, contextos e possíveis microsserviços

## Etapa 6 - Construindo uma Lista de Microserviços

### - Se esta for a Arquitetura Escolhida

### (Software Design)

*Tempo*

De 60 a 90  
Minutos

**Objetivo:** Reclassificação e resultado da Modelagem de Software.

#### Quais são os possíveis microsserviços que podem atender nosso fluxo de negócio?

Agora com a equipe técnica, devemos tentar listar os microsserviços necessários para cada contexto delimitado.

Um contexto delimitado não é a mesma coisa que um microsserviço, podendo haver vários microsserviços correspondentes a um agregado.

Em sistemas simples, isso pode ser feito através de um mapeamento 1-1.

Em contextos mais complexos, pode ser feito por agregado contido em um contexto delimitado.

Premissas a ser observadas na definição de microsserviços:

» Devemos tentar eliminar a necessidade de solicitações síncronas entre os serviços;

» Devemos evitar pontos únicos de falha (SPOF - Single Point of Failure). Por exemplo, uma arquitetura em que todos os serviços precisam chamar o microsserviço de pedido para concluir seus processos);

🔗 Se uma entidade ou agregado aparecer em dois contextos, isso significa que devem haver dois microsserviços independentes para lidar com os processos de negócio divergentes nesses dois contextos;

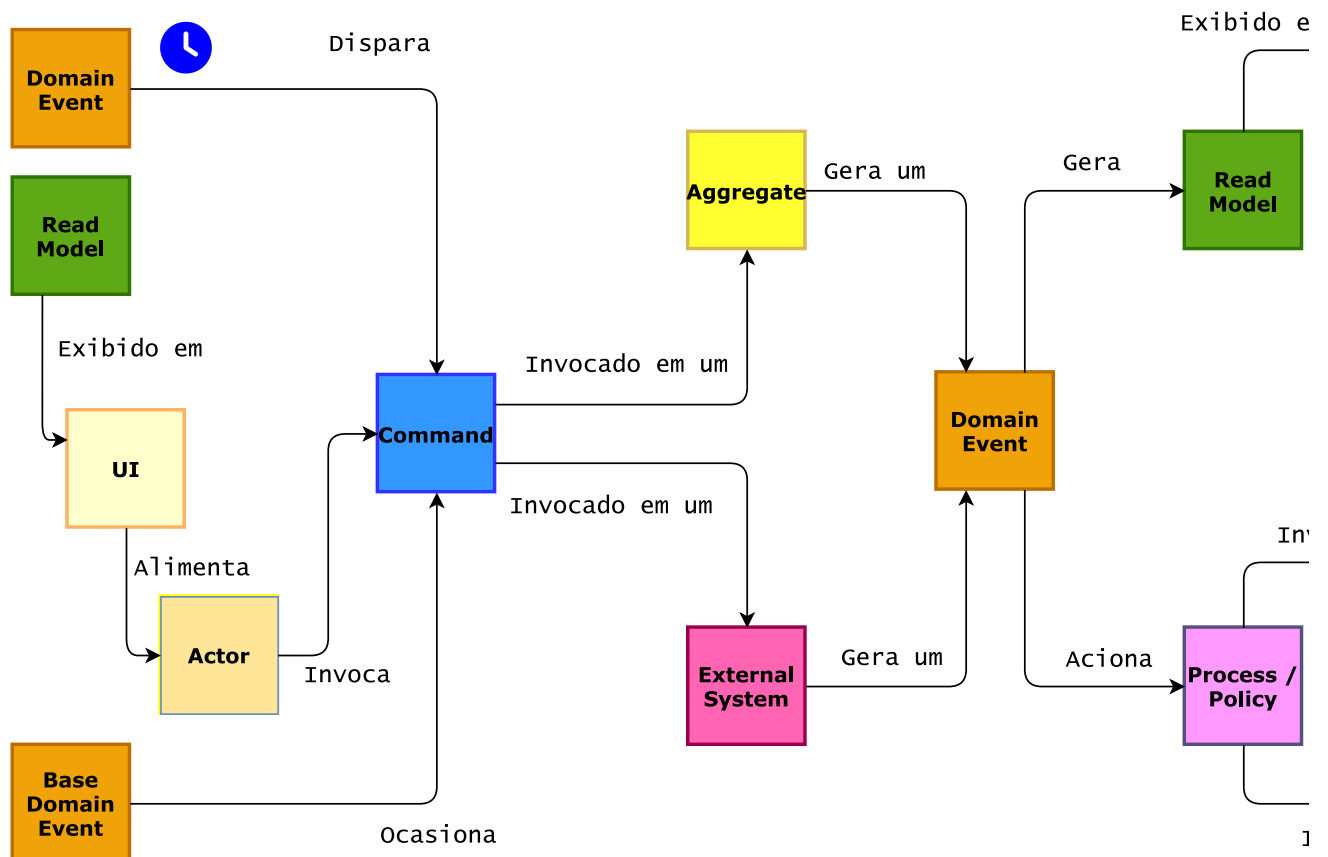
🔗 Se um microsserviço estiver envolvido em vários contextos delimitados, o alarme deve soar! Isso geralmente resulta em um monólito distribuído que está fortemente acoplado.

**Elementos usados nesta etapa:**

### Microsserviços Condicionados

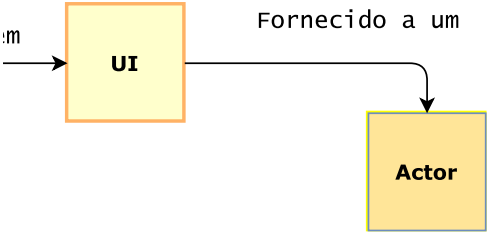
- 1) Pedido
- 2) Estoque
- 3) Crédito
- ...

## Fluxo de Controle





NG



Hot Spot

